

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra Měřicí a řídicí techniky

**Lokalizační a komunikační systém pro mobilní
robotické zařízení**

**Localisation and Communication System for Mobile
Robotic Device**

Zadání diplomové práce

Student: **Bc. Hynek Prokop**
Studijní program: N2649 Elektrotechnika
Studijní obor: 2601T004 Měřicí a řídicí technika
Téma: **Lokalizační a komunikační systém pro mobilní robotické zařízení**
Localisation and Communication System for Mobile Robotic Device

Zásady pro vypracování:

1. Návrh senzorického systému pro orientaci zařízení v prostoru.
2. Snímače polohy a prostoru.
3. Výběr vhodné platformy (HCS12 nebo ARM Freescale iMX31 LiteKit.).
4. Návrh snímačového modulu.
5. Výběr vhodného OS (Free RTOS, Linux, QNX, Win CE, ...).
6. Výběr vhodného způsobu komunikace mezi elektronickými moduly v zařízení.
7. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

1. BARRY, Richard. *Using the FreeRTOS real time Kernel : A Practical Guide*. [s.l.] : [s.n.], 2009. 164 s.
2. BARRY, Richard. *FreeRTOS Reference manual : API Functions and Configuration options*. [s.l.] : [s.n.], 2009. 119 s.
3. CAN in Automation. *CANopen Application Layer and Communication Profile : CiA Draft Standard 301*. [s.l.] : [s.n.], 2002. 135 s.
4. BORENSTEIN, L., EVERETT, H. R., FENG, L. *Where am I? : Sensors and Methods for Mobile Robot Positioning*. J. Borenstein. [s.l.] : [s.n.], 1996. 282 s.
5. Motorola, Inc. *MC9S12DP256B : Device User Guide. 2nd edition*. [s.l.] : [s.n.], 2005. 126 s.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

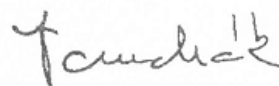
Vedoucí diplomové práce: **Ing. Jiří Kotzian, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Ing. Jiří Koziolek, Ph.D.
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Bc. Hynek Prokop

Datum odevzdání diplomové práce: 7. 5. 2010

Poděkování

Chtěl bych touto cestou velmi poděkovat vedoucímu mé diplomové práce panu Ing. Jiřímu Kotzianovi, Ph.D. za cenné rady, konzultace a připomínky spojené s vypracováním mého úkolu. Dále pak pánům Marianu Kurucovi, Tomáši Lippovi a Jaromíru Konečnému, kteří se podíleli na realizaci vozidla.

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací lokalizačního a komunikačního systému autonomního robotického zařízení. Zpracovává danou problematiku od základů. Začíná výběrem komponent, návrhem a sestavením desek plošných spojů a využitím operačního systému FreeRTOS pro mikroprocesorové systémy. Pokračuje způsoby lokalizace robotického zařízení v budovách a venkovních prostředích, například systémem GPS nebo měřením odometrie. Zaobírá se také výběrem a implementací komunikačního rozhraní mezi moduly zařízení, které je nakonec vytvořeno sběrnici CAN s aplikační vrstvou CANopen.

Klíčová slova

Sběrnice CAN, mikroprocesor freescale řady HCS12, robotické zařízení, lokalizační systém GPS, CodeWarrior, CANopen, Formica, FreeRTOS, odometrie, lokalizace v prostoru, komunikace.

Abstract

This thesis describes the design and implementation of localization and communication system of an autonomous robotic device. And handle the issue from scratch. It begins with the selection of components, design and assembly of printed circuit boards and using the operating system FreeRTOS for microprocessor systems. Then continue with localization issues of robotic device in buildings and outdoor environments, such as GPS or odometry measurements. Deals with the selection and implementation of communication interfaces between modules of device that is ultimately created by CAN bus with the CANopen application layer.

Keywords

CAN bus, Freescale microprocessor HCS12, robotic device, localization system GPS, CodeWarrior, CANopen, Formica, FreeRTOS, odometry, localization, communication.

Seznam použitých symbolů a zkratek

ACK	- (Acknowledge) Potvrzení
AD	- Analogově digitální převodník
ADC	- (Analog to Digital Converter) Analogově digitální převodník
API	- (Application Programming Interface) Rozhraní pro programování aplikací
ASCII	- (American Standard Code for Information Interchange) Americký standardní kód pro výměnu informací
baud	- Jednotka modulační rychlosti
CAL	- (CAN Application Layer)
CAN	- (Controller Area Network) Datová sběrnice
CANopen	- Aplikační vrstva na sběrnici CAN
CMOS	- Technologie používaná na převážnou většinu integrovaných obvodů.
COB-ID	- (Communication Object Identifier) Označení identifikátoru zprávy dle specifikace CANopen
CRC	- (Cyclic redundancy check) Cyklický redundantní součet
CSMA/CD	- (Carrier Sense Multiple Access with Collision Detection) Protokol pro přístup k přenosovému médium
DAM MPDO	- (Destination Address Mode PDO) Označení multiplexované PDO zprávy
DC	- (Direct Current) Stejnosměrný proud
DGPS	- Technologie pro zpřesnění GPS
DLC	- Označení části zprávy sběrnice CAN nesoucí informaci o délce zprávy
DME	- (Distance Measuring Equipment) Impulzní dálkoměr
DPS	- Deska plošných spojů
ECT	- (Enhanced Capture Timer) Časovač/Čítač procesorů rodiny HCS12
EEPROM	- (Electrically Erasable Programmable Read-Only Memory) Elektricky mazatelná nevolatilní paměť
EMCY	- (Emergency) Označení protokolu (CANopen)
EOF	- (End of frame) Ukončovací pole zprávy sběrnice CAN
FIFO	- Paměť typu First In First Out
GGA	- (Global Positioning System Fixed Data) Označení zprávy dle specifikace NMEA
GND	- Napájecí zem
GNU	- Verze GPL licence
GPS	- Global Positioning System
HCS12	- Označení výrobní řady mikroprocesorů firmy Freescale
HDOP	- (Horizontal Dilution of Precision) Přesnost určení polohy v horizontálním směru
ID	- (Identifier) Identifikátor
IDE	- (Identifier Extended) Rozšířený identifikátor na sběrnici CAN
IIC	- (Inter-Integrated Circuit) Označení komunikační sběrnice
ILS	- (Instrument landing system) Určování polohy letadla při konečném přiblížení
IR	- (Infrared) Druh elektromagnetického záření
ISO/OSI	- Referenční model
LCD	- (Liquid crystal display) Displej z tekutých krystalů
LED	- (Light Emitting Diode) Elektronická součástka emitující světlo
Mbps	- (Millions of Bits Per Second) Jednotka rychlosti datového přenosu
MCLR	- (Master Clear) Pin využívaný pro přivedení programovacího napětí
MSCAN	- (Motorola Scalable Controller Area Network) Komunikační rozhraní zabudované v procesorech Freescale
N/A	- (Not Applicable or Not Available) Označení dat, která nemohou být poskytnuta

NC	- (No Connection) Žádné spojení (nepřipojovat)
NDB	- (Non Direction Beacon) Rádiový kompas
NMEA	- (National Marine Electronics Association) komunikační standard pro komunikaci se zařízeními využívajícími GPS
NMT	- (Network Management) Zpráva definovaná standardem CANopen
Node-ID	- Identifikátor zařízení na sběrnici
OD	- (Object Dictionary) Slovník objektů
PDO	- (Proces Data Objects) Zpráva definovaná standardem CANopen
PSD	- Označení principu polohové citlivého senzoru
PWM	- (Pulse Width Modulation) Pulzní šířková modulace
QFN	- (Quad Flat No leads) Označení pouzdra integrovaných obvodů
RAM	- (Random Access Memory) Označení typu paměti
$R_{DS(ON)}$	- Odpor unipolárních tranzistorů v sepnutém stavu
RMC	- (Recommended Minimum Specific GNSS Data) Označení zprávy dle specifikace NMEA
RPDO	- (Receive Proces Data Objects) Zpráva definovaná standardem CANopen
RS232	- Označení komunikačního rozhraní
RSMA	- (Reverse polarity SMA) Označení konektoru
$R_{th(j-a)}$	- Tepelný odpor - přechod - okolí
RTR	- (Remote transmission request) Žádost o vyslání zprávy na sběrnici CAN
r_z	- Poloměr Země
SAM MPDO	- (Source Adress Mode PDO) Označení multiplexované PDO zprávy
SCI	- (Serial Communications Interface) Komunikační rozhraní zabudované v procesorech freescale
SCL	- Označení vodiče hodinového signálu sběrnice IIC
SDA	- Označení datového vodiče sběrnice IIC
SDO	- (Service Data Object) Zpráva definovaná standardem CANopen
SiFR	- Komunikační protokol používaný v GPS zařízeních
SIL	- (Safety Integrity Level) Označení stupně bezpečnosti
SOF	- (Start of frame) Zahajovací pole zprávy sběrnice CAN
SPI	- (Serial Peripheral Interface) Označení komunikační sběrnice
SYNC	- (Synchronization) Zpráva definovaná standardem CANopen
t_D	- Časový krok PWM signálu pro termočlávkové pole
t_H	- Šířka impulzu PWM signálu
TPDO	- (Transmit Proces Data Objects) Zpráva definovaná standardem CANopen
TTL	- (Transistor Transistor Logic) Technologie používaná k výrobě integrovaných obvodů
u_a	- Teplota okolí
U_{BE}	- Napětí na přechodu báze – emitor bipolárního tranzistoru
U_H	- Výstupní napětí TTL logiky při logické jedničce.
u_j	- Teplota přechodu polovodičové součástky
USB	- (Universal Serial Bus) Označení komunikační sběrnice
UTC	- (Coordinated Universal Time) Koordinovaný světový čas
V_{cc}	- Napájecí napětí
VOR	- (VHF Omnidirectional Radio Range) Kurz přijímače k majáku
WGS-84	- Eliptický souřadnicový systém, který používá zařízení GPS
WiFi	- Označení pro lokální bezdrátové síť
xor	- Logická operace

Obsah

1	ÚVOD	1
2	ANALÝZA SOUČASNÉHO STAVU	2
3	CÍLE	3
4	NÁVRH ŘEŠENÍ	4
4.1	PRŮZKUMNÉ VOZIDLO	4
4.2	MODUL POLOHY	6
4.3	MODUL SENZORŮ	6
5	SENZORICKÝ SYSTÉM	7
5.1	MOŽNOSTI URČOVÁNÍ POLOHY	7
5.1.1	Předem známé údaje	7
5.1.2	Relativní údaje o poloze	7
5.1.3	Absolutní údaje o poloze	8
5.2	SNÍMAČE PRO URČOVÁNÍ POLOHY	10
5.3	SNÍMAČE VZDÁLENOSTI	14
5.4	SNÍMAČE OKOLNÍHO PROSTŘEDÍ	16
6	NÁVRH MODULŮ	20
6.1	VÝBĚR VHODNÉ PLATFORMY	20
6.2	NÁVRH MODULU SNÍMAČŮ	21
6.2.1	Spínaný zdroj s měřením proudu	22
6.2.2	Zapojení sběrnice CAN	22
6.2.3	Zapojení unipolárních tranzistorů	23
6.2.4	Připojení inkrementálních čidel	24
6.2.5	Rozložení desky plošných spojů	24
6.3	NÁVRH MODULU POLOHY	25
6.3.1	Zapojení GPS přijímače	25
6.3.2	Akcelerometr	26
6.3.3	Senzor tlaku a senzor koncentrace výbušného plynu	27
6.3.4	Rozložení desky plošných spojů	27
7	OPERAČNÍ SYSTÉM	28
7.1	VÝBĚR VHODNÉHO OPERAČNÍHO SYSTÉMU	28

7.2	FREERTOS	29
7.2.1	Práce s vlákny – plánovač	29
7.3	ÚPRAVA FREERTOS PRO PROCESOR MC9S12DP512	30
7.3.1	Vytvoření nového projektu	31
7.3.2	Generace časových přerušení pro plánovač	32
7.3.3	Algoritmus spouštění operačního systému	34
7.3.4	Měření provedená na operačním systému	35
8	KOMUNIKACE MEZI MODULY NA VOZIDLE	36
8.1	VÝBĚR TYPU KOMUNIKACE	36
8.1.1	Sběrnice CAN	37
8.1.2	Implementace sběrnice CAN	38
8.2	VÝBĚR VYŠŠÍ VRSTVY SBĚRNICE CAN	39
8.2.1	CANopen	40
8.2.2	Implementace CANopen	41
8.3	KOMUNIKACE S HLAVNÍ ŘÍDICÍ JEDNOTKOU	45
8.3.1	Výběr komunikace	45
8.3.2	Implementace rutin pro příjem a vysílání	45
9	URČOVÁNÍ POLOHY V PROSTORU	46
9.1	MAPOVÉ PODKLADY	46
9.2	ODOMETRIE	46
9.3	ELEKTRONICKÝ KOMPAS	50
9.4	NAVIGAČNÍ SYSTÉM GPS	51
9.5	KOMBINACE DOSTUPNÝCH ÚDAJŮ O POLOZE	53
10	POPIS SOFTWARE	54
10.1	MODUL POLOHY	54
10.2	MODUL SENZORŮ	58
11	ZÁVĚR	61
	LITERATURA	63
	SEZNAM PŘÍLOH	65

1 Úvod

Existují situace a místa, které mohou být pro pobyt nebo práci člověka příliš nebezpečné. Od nepaměti se proto konstruuují přístroje a zařízení, které jsou schopny alespoň zčásti nahradit jeho přítomnost. Zprvu se jednalo o víceméně jednoduchá zařízení ovládaná na dálku, ale nyní je celosvětovým trendem vytvořit zařízení, které i bez lidské pomoci dokáže prozkoumávat, nebo i pracovat v prostředí, pro člověka neznámém nebo dokonce nebezpečném.

Aby takovéto zařízení bylo autonomní, je důležité, aby dokázalo určit svou polohu. Protože pokud ji určit nedokáže, je téměř nemožné se správně rozhodovat, kterým směrem se bude pohybovat dál. K určení své polohy může získávat relativní a absolutní údaje o poloze. Ne všechny senzory a měření jsou dokonalá a vnášejí do výsledků různě velké chyby. Vhodnou kombinací těchto údajů můžeme tyto chyby částečně eliminovat.

Velké nároky na funkčnost sebou jistě přináší i komplikovanější řešení. Je výhodné takovéto řešení rozložit na více relativně soběstačných a jednodušších částí a ty pak propojit. Z toho plyne požadavek na volbu vhodného komunikačního rozhraní a samozřejmě také komunikačního protokolu.

Tato práce je založena na průzkumném vozidle, které bylo vytvořeno v rámci prací bakalářských. Toto vozidlo, jeho přednosti, ale hlavně nedostatky jsou popsány v následující kapitole 2. Vymezení oblastí, do kterých tato práce spadá, je popsáno v kapitole cílů s číslem 3. Následuje kapitola 4, která čtenáři přibližuje nejen návrh řešení zadané problematiky, ale také mu nastiňuje robotické zařízení jako celek.

Kapitola nazvaná „Senzorický systém“ (Kap. 5) jednak teoreticky rozebírá a třídí možnosti určování polohy, ale také vyjmenovává vybrané senzory, které budou na zařízení použity. Ve zkratce popisuje jejich základní vlastnosti a možnosti připojení k aplikaci.

Návrh modulů, jejich zapojení i jejich desky plošných spojů, je popsán v kapitole 6.

Kapitola nazvaná „Operační systém“ (Kap. 7) zprvu krátce popisuje vybraný operační systém reálného času a následně se zabývá jeho přizpůsobením pro daný hardware. Na konci této kapitoly se nachází zajímavá část popisující nároky tohoto OS na výpočetní čas mikroprocesoru.

Důležitou kapitolou (Kap. 8) této práce je část zabývající se výběrem a implementací komunikace mezi elektronickými moduly v zařízení. Jen zběžně se zde popisují nutné teoretické základy a větší část se zabývá nastíněním způsobu, jak je s využitím API funkcí operačního systému komunikace implementována.

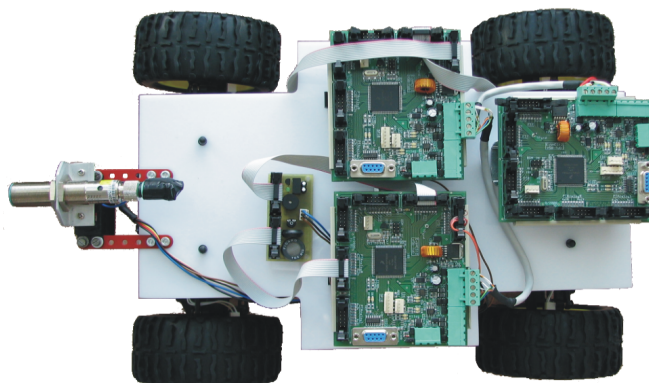
Kapitola zabývající se určováním polohy v prostoru (Kap. 9) popisuje, jakých metod určování polohy se využívá právě na tomto zařízení. Tyto jednotlivé metody jsou rozebírány i z hlediska jejich implementace.

Neméně důležitou částí je kapitola (Kap. 10) popisující softwarové vybavení jednotlivých modulů. Jsou v ní podrobněji popsány rutiny a vlákna pro měření a obsluhu jednotlivých senzorů a aktuátorů.

Dosažené výsledky a návrhy pro další práci popisuje poslední kapitola 11.

2 Analýza současného stavu

Tato práce částečně vychází z průzkumného vozidla, které vzniklo v rámci bakalářských prací. To je postaveno na podvozku malého RC modelu, na kterém jsou osazeny tři moduly s procesory architektury Freescale HCS12. Jeden z nich se stará o pohon, druhý o senzory a třetí o určování polohy, výběr trasy a bezdrátovou komunikaci s PC. Moduly jsou propojeny sběrnici CAN s vlastním (ne-standardním) protokolem.



Obr. 1 Fotografie vozidla, ze kterého se vychází

Pro měření okolního terénu je na vozidle umístěn ultrazvukový senzor vzdálenosti, který je uchycen na servo-pohonu. Díky jeho malé frekvenci měření a malému dosahu (1metr) se musí vozidlo pohybovat velice pomalu. Případné určení polohy překážky je problematické z důvodu malého počtu změřených bodů.

Vozidlo zjišťuje svou polohu pouze ze systému GPS. Není v něm navržen žádný souřadnicový systém, neobsahuje mapy a neimplementuje žádné sofistikované algoritmy plánování tras, nebo vyhýbání se překážkám. Komunikace s PC probíhá přes GSM síť. Její výhodou je téměř všudypřítomné pokrytí, ale zařízení potřebuje SIM karty a připojování přes operátora je složité a zdlouhavé.

Pohon vozidla je navrhnut na vyšší rychlosti a kvalita regulace při pomalé jízdě není optimální.

3 Cíle

Společným cílem týmu lidí je navrhnout a sestavit robotické zařízení schopné se autonomně pohybovat ve zmapovaném prostředí. Dokáže se v prostoru orientovat a bezdrátově komunikovat s vizualizační stanicí, které poskytuje různé informace o své poloze. Každé robotické zařízení by mělo informovat operátora o okolí, ve kterém se pohybuje. Proto bude navržen senzorický systém nejen pro lokalizaci, ale také pro monitorování okolního prostředí. Určitě na zařízení nebudou chybět například senzory teploty nebo tlaku.

Tato práce se převážně zabývá lokalizačními a komunikačními systémy takového zařízení. Je nutné vybrat takové snímače polohy a prostoru aby byl lokalizační systém schopný fungovat jak uvnitř budov, tak ve venkovních prostředích. Aby údaje o poloze byly srozumitelné i pro hlavní řídicí jednotku, je nutné vytvořit jednotný souřadnicový systém, ve kterém bude možné určit polohu a natočení zařízení.

S největší pravděpodobností se bude jednat o distribuovaný řídicí systém sestavený z několika modulů. Tyto moduly budou propojeny vybraným komunikačním rozhraním s vybraným protokolem. Komunikační protokol bude vytvořen tak, aby se dal snadno modifikovat pro případné další funkce robotického zařízení.

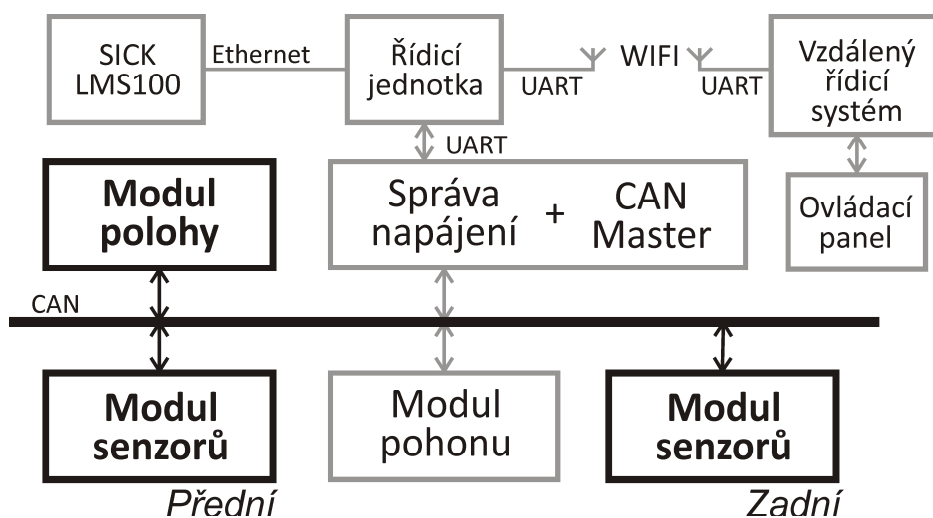
Při vytváření funkčního zařízení, je také nezbytné navrhnout a sestavit hardware jednotlivých modulů. Je proto nutné sestavit schémata zapojení, vytvořit desky plošných spojů a zajistit jejich výrobu ve specializované firmě.

Při implementaci řešení je nutné zvážit využití některého z dostupných operačních systémů pro mikroprocesorové systémy. S tím souvisí i zhodnocení náročnosti vybraného OS na výpočetní čas a porovnání jeho přínosů a omezení spojených s jeho implementací.

4 Návrh řešení

4.1 Průzkumné vozidlo

Řízení průzkumného vozidla je koncipováno jako distribuovaný řídicí systém. Ten je složen z osmi modulů, sedm z nich je umístěno přímo na vozidle a komunikují spolu prostřednictvím sběrnice CAN, sériového portu a sběrnice ethernet. Osmou část tvoří vzdálený systém s vizualizací, který s vozidlem komunikuje prostřednictvím sítě Wi-Fi a slouží pro sledování vozidla a jeho manuální ovládání. Na obrázku Obr. 2 je zobrazeno blokové schéma návrhu řešení, rozčlenění palubní elektroniky dle jejich funkce na jednotlivé moduly.

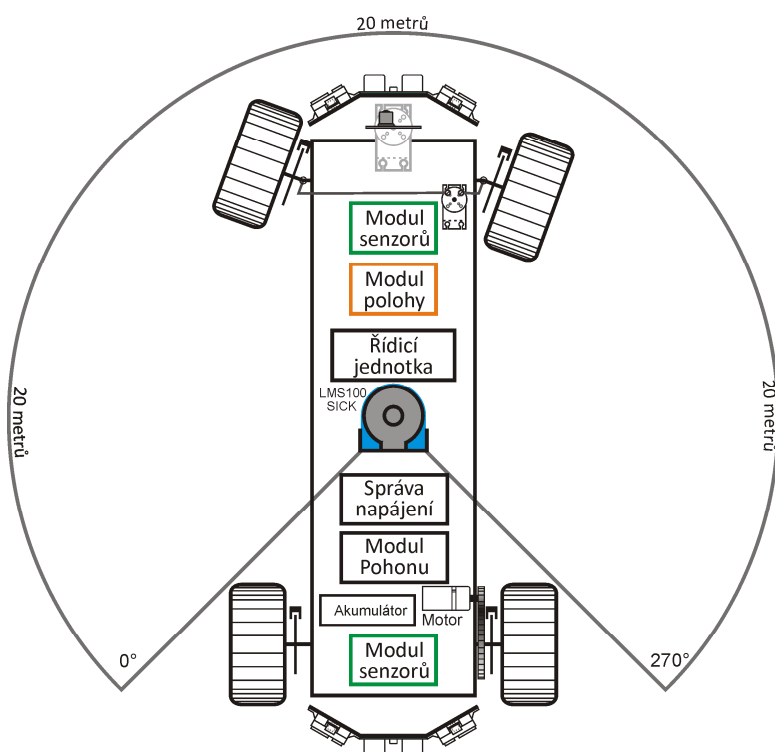


Obr. 2 Blokové schéma vozidla

Popis jednotlivých modulů:

- **Řídicí jednotka** - Jejím úkolem je navázání spojení se vzdáleným ovládacím panelem, výběr a výpočet nejvhodnější trasy k zadanému cíli, vyhodnocení překážek před a za vozidlem. Prostřednictvím sběrnice ethernet je k jednotce připojen laserový scanner SICK LMS100 pro měření vzdálenosti překážek kolem vozidla. Řízení vozidla se bude provádět prostřednictvím zadávání příkazů ostatním (podřazeným) modulům.
- **Správa napájení** – Slouží pro distribuci elektrické energie z akumulátoru do jednotlivých modulů na vozidle. Řídí nabíjení, dobíjení a sleduje stav baterie při vybíjení a dále jí vyhodnocuje. Je také datovým mostem mezi CAN sběrnici a nadřazeným systémem.
- **Modul polohy** - Poskytuje řídicí jednotce informace o aktuální pozici vozidla. Tu vypočítává z globálních a relativních informací o poloze z připojených senzorů. Mimo to získává informace o okolním prostředí jako je teplota, tlak, koncentrace výbušných látek.

- **Modul senzorů** – Na vozidle se nacházejí dva tyto moduly. Ty na vozidle fungují jako sběrné uzly, které shromažďují informace ze svého okolí a ty pak poskytují ostatním modulům. Modul nacházející se na přední části vozidla zároveň ovládá natočení předních kol.
- **Modul pohonu** – Obsahuje výkonovou část pro řízení pohonu. Implementuje algoritmus pro regulaci otáček motoru, proudové omezení atd. Požadovaná rychlost pojezdu je ovládána prostřednictvím příkazů z nadřazeného systému.
- **Vzdálený řídicí systém** - Vizualizace a správa vozidla. Data z vozidla jsou přenášena prostřednictvím sítě Wi-Fi a použita pro kontrolu, zadávání nových parametrů a jejich vizualizaci na dotykovém LCD displeji.



Obr. 3 Koncepte průzkumného vozidla

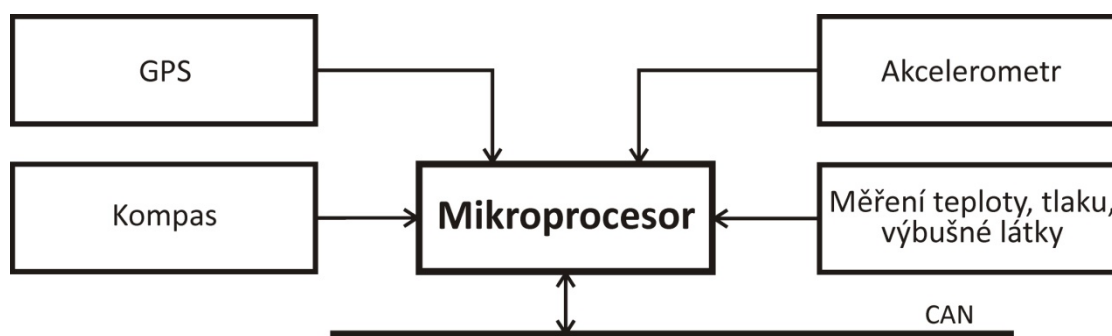
Na obrázku Obr. 3 je zobrazeno navržené rozmístění komponent na průzkumném vozidle. Moduly jsou umístěny tak, aby se omezila délka signálových vedení k jejich senzorům a aktuátorům.

Moduly senzorů a modul polohy, které jsou na obrázku Obr. 3 zvýrazněny, budou dále popsány v následující kapitole.

4.2 Modul polohy

Modul polohy, kombinací relativních a absolutních údajů o poloze, bude udržovat a aktualizovat informaci, kde se vozidlo nachází a jakým směrem je natočeno. Modul bude obsahovat přijímač signálu GPS, elektronický kompas a akcelerometr k měření náklonu vozidla. Dále bude zjišťovat informace o svém okolním prostředí jako teplotu, tlak a koncentraci výbušných látek ve vzduchu. Modul bude napojen na komunikační sběrnici CAN.

Modul polohy spolupracuje s moduly senzorů, které mu poskytují relativní údaje o změně polohy, kterou měří z ujeté dráhy vozidla.

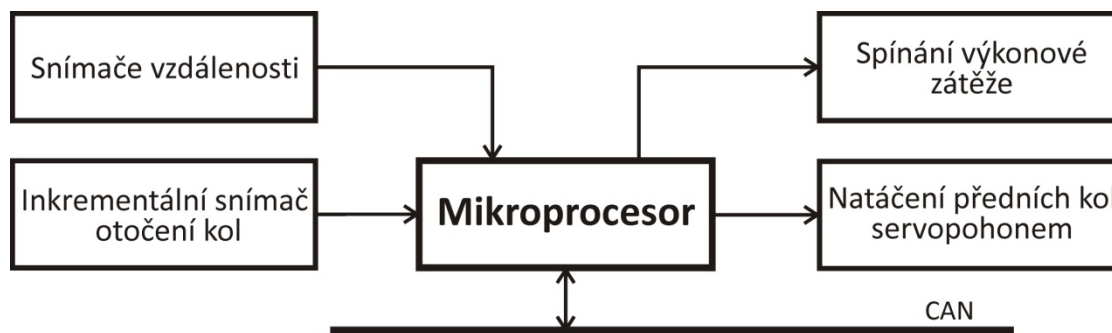


Obr. 4 Blokové schéma modulu polohy

4.3 Modul senzorů

Modul senzorů bude fungovat jako sběrný uzel. Bude shromažďovat informace ze senzorů a odesílat je do hlavní řídicí jednotky. Informace týkající se o pohybu vozidla bude poskytovat i modulu polohy. Jednotka bude schopna měřit vzdálenost vozidla od překážek, ujetou dráhu vozidla. A dále bude spínat různé spotřebiče (například světla) a servopohonem řídit natočení kol. Modul bude napojen na komunikační sběrnici CAN.

Na vozidle budou umístěny dva tyto moduly (Obr. 3). Jednotka umístěná v zadní části bude prostřednictvím inkrementálních čidel měřit ujetou dráhu a bude ji přeposílat do modulu polohy. Modul v přední části měří a posílá totéž, a zároveň ovládá natočení předních kol vozidla.



Obr. 5 Blokové schéma modulu senzorů

5 Senzorický systém

Jestliže je vyžadováno, aby se robotické zařízení pohybovalo ze zadaného bodu do druhého, je nutné určit polohu startu, cíle a samozřejmě neustále aktualizovat polohu samotného zařízení. Důležitou informací je také údaj o natočení zařízení. To vše musí být umístěno v určitém souřadném systému.

Vše se ještě více komplikuje v případě, kdy si má zařízení určit svou polohu samo i po zapnutí. Obdobná situace nastává i při takzvaném „únosu“ zařízení, kdy se během provozu přemístí na neznámé místo. Další komplikací při určování polohy je dynamicky měnící se prostředí, čímž se značně komplikuje použití map.

V tomto případě se omezíme na statické prostředí, nepředpokládáme „únos“ zařízení, a při jeho zapnutí se mu zadá jeho počáteční poloha a hodnota natočení.

5.1 Možnosti určování polohy

Dostupné informace pro určení polohy se dají rozdělit do tří skupin. První skupinu tvoří takové informace, které jsou předem známé, jako jsou například mapové podklady prostředí, ve kterém se zařízení bude pohybovat. Druhou skupinu tvoří relativní údaje o změně polohy. V případě, kdy je známa počáteční poloha, dokážeme z nich určit aktuální souřadnice sečtením relativního a počátečního údaje. Jejich nevýhodou je zvyšující se nepřesnost s ujetou vzdáleností. Třetí skupinou jsou absolutní údaje. Ty sice nemají takovou absolutní přesnost, ale na druhou stranu se tato přesnost radikálně nemění s časem.

5.1.1 Předem známé údaje

Tyto údaje si ve většině případů robotické zařízení vozí sebou, a jsou do něj vložena před jízdou. Dobrým příkladem jsou mapové podklady. V nich jsou zakreslené jednotlivé překážky, nebo dokonce jsou vyznačeny cesty, kterými se zařízení může pohybovat. V mapách také mohou být rozlišeny statické a proměnné překážky. Příkladem mohou být dveře, ty mohou, ale nemusí být otevřeny.

5.1.2 Relativní údaje o poloze

Zjišťování aktuální polohy sečtením uražené dráhy v určitém směru a poslední známe polohy je metoda, která se používá snad již od dob, kdy lidstvo začalo cestovat. Této metody se využívalo, dokonce v některých případech se využívá i dnes, v námořní a letecké dopravě. Stačí pouze měřit rychlost a čas cesty. A z těchto údajů vypočítat dráhu a tu přičíst k poslední známe poloze. Zřejmou nevýhodou je akumulace nepřesností, způsobených nepřesným měřením dráhy, nebo směru pohybu. Někdy se taky označuje jako metoda hrubého určení polohy (dead reckoning).

V dnešních aplikacích se pro relativní snímání změn polohy používá odometrie, nebo inerciální snímače.

Odometrie

Většina robotických zařízení se pohybuje s pomocí kol, pásů nebo nohou. Díky znalostem o dráze uražené po otočení jedné otáčky kola, nebo po jednom kroku určitým směrem, můžeme odhadovat, o kolik se změnila poloha zařízení.

Nejčastěji se vyskytují zařízení s koly, které pomocí inkrementálních snímačů zjišťují ujetou dráhu určitým směrem. Takto zjištěná změna polohy je přesná jen pro krátké vzdálenosti. Velikost chyby závisí na povrchu, po kterém se zařízení pohybuje, rychlosti jakou se pohybuje, a také na přesnosti udržení směru pohybu zařízení (mechanické vůle geometrie kol). V potaz se musí brát také to, že odometrie měří skutečnou ujetou vzdálenost a ne posun ve dvojrozměrném prostoru. Příkladem může být strmý kopec, který musíme překonat. Změřená relativní změna polohy bude mnohem větší, než skutečná vzdálenost startu a cíle.

I když je odometrie pro větší vzdálenosti nepřesná, je často využívána díky své nízké odezvě měření a snadné realizovatelnosti.

Inerciální snímání změn polohy

Další metodou hrubého určení polohy využívá inerciálních snímačů (akcelerometry a gyroskopy). Měření je založeno na Newtonově pohybovém zákonu, který říká, že vyprodukovaná síla je přímo úměrná zrychlení tělesa. Dle dané závislosti je možné vypočíst změnu rychlosti a pozice v čase. Zrychlení je zaznamenáváno akcelerometry ve třech osách, ze kterých ve spojení s gyroskopy určující náklon zařízení vůči souřadnému systému, určíme uraženou dráhu.

Přesnost určení polohy závisí na povrchu, po kterém se vozidlo pohybuje. Čím větší jsou nerovnosti, tím více dochází k růstu akumulované chyby.

Tento typ určování polohy byl zaveden v letecké dopravě, ze které se rozšířil i do dalších odvětví. Komerčně dostupné verze inerciálních přístrojů pro určování polohy dosahují přesnosti pod 0.1%, ale jsou zároveň velmi drahé.

5.1.3 Absolutní údaje o poloze

Absolutní údaje mají tu výhodu, že údaj o poloze zařízení získáme jedním měřením. Přesnost takto zjištěného údaje může být různá (od kilometrů až po milimetry), a záleží na použité technologii. Výhodou je pak určitá jistota, že na rozdíl od údajů relativních zůstává přesnost určení polohy s časem konstantní.

Objekty v prostoru (LandMarks)

Tato metoda lokalizace je založena na detekci objektů v prostoru, podle kterých je zařízení schopno určit svou polohu. Tyto objekty můžeme rozdělit na aktivní a pasivní. Pasivní objekty se pak dále dělí na umělé nebo přirozené.

Aktivní objekty do svého okolí vysílají signály (rádiové, ultrazvukové, infračervené, chemické), které robotické zařízení přijímá a vyhodnocuje. Nejjednodušeji získanou informací je

samotná existence takového signálu, která i přes velkou nepřesnost, určuje polohu v prostoru. Další možností určování polohy je využití triangulace ze signálů z více zdrojů se známou polohou. Příkladem může být systém GPS. Po mnoho let se již také používají metody radionavigace, které se převážně vyskytují v leteckém provozu. Za zmínku stojí systémy NDB (rádiový kompas), DME (impulzní dálkoměr), VOR (kurz přijímače k majáku) a ILS (určování polohy letadla při konečném přiblížení k letišti).

Pasivní objekty jsou takové, které nic nevysílají, a spoléhá se na jejich detekci samotným zařízením. Objekty, které se do prostředí umísťují záměrně, patří do skupiny takzvaných pasivních objektů umělých. Příkladem mohou být čárové kódy umístěné na podlaze v budově, které nesou informaci o své poloze. Objekty vyskytující se v prostředí běžně, nazýváme přirozené. Příkladem mohou být stromy, budovy nebo cesty.

Metoda určování polohy detekcí pasivních objektů v okolí se podobá způsobu, kterým se v prostoru orientuje člověk. Zařízení musí obsahovat senzory schopné takovéto objekty zachytit. Mohou to být kamery, senzory vzdálenosti a jiné.

Určování polohy z mapy

Pokud zařízení obsahuje mapu prostředí, ve kterém se pohybuje, je možné díky ní a různým senzorům určovat jak natočení, tak i vlastní polohu. V mapách sice nemůžou být zaznamenány všechny detaily (židle, stoly, nábytek), ale například dveře, stěny, chodby je možné relativně jednoduše rozpoznat. Podmínkou ale je, že zařízení má alespoň nějakou informaci o tom, ve kterém místě budovy se nachází.

Existuje také možnost, že si zařízení mapu vytváří průběžně při prozkoumávání okolí. V tom případě ale potřebuje přesnější údaje o své poloze, protože zanesení překážek do mapy s nesprávným umístěním se nám v budoucnu „odmění“ nepřesným určením polohy.

Měření magnetického pole země

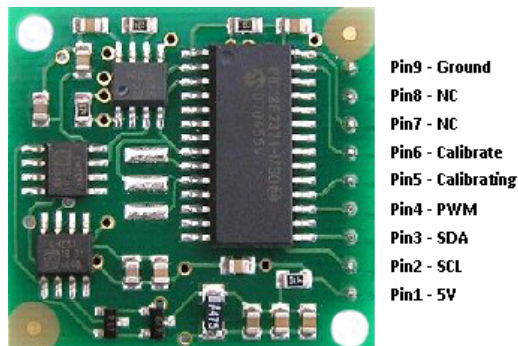
Využití magnetického pole země k určování světových stran se využívá již od čtvrtého století našeho letopočtu. V dnešní době existuje mnoho přístrojů pro určování azimutu pracujících na různých principech. Problém tkví v nehomogenním magnetickém poli, v rušeních způsobeném různými přístroji a v železobetonové konstrukci budov. V některých případech se může azimut měřený v budovách lišit i o desítky stupňů.

[7] [11] [13]

5.2 Snímače pro určování polohy

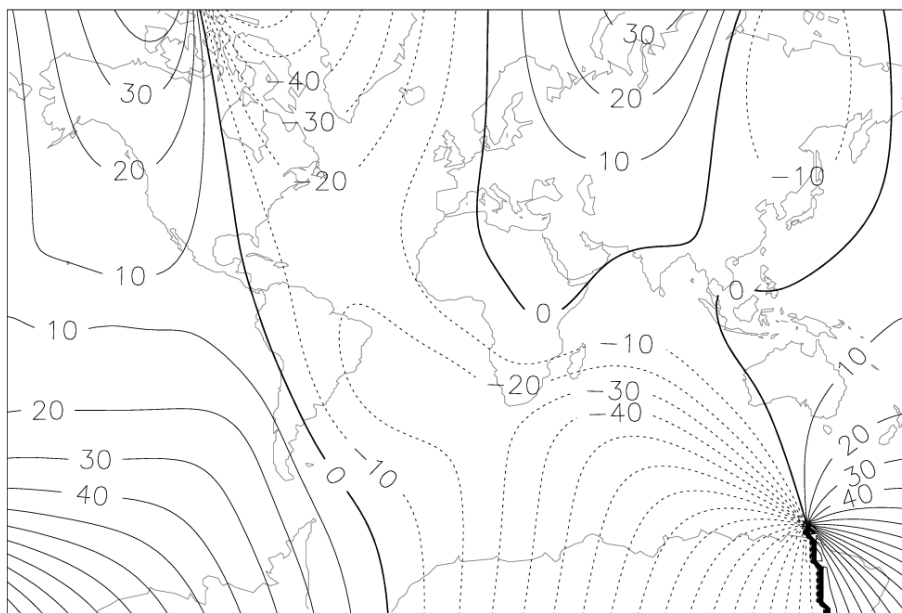
Elektronický kompas

Kompas je koupen jako hotový modul, se kterým se komunikuje prostřednictvím sběrnice IIC. Je osazen senzory magnetického pole Philips KMZ51, které mají dostatečnou citlivost i pro zachycení magnetického pole země. Osazený procesor vypočítává směr magnetického pole země ze dvou senzorů pootočených o úhel 90° .



Obr. 6 Elektronický kompas CMPS03

Jelikož magnetické pole není stejnorodé a souběžné s poledníky země, obsahuje elektronický kompas možnost „ruční“ kalibrace odchylky magnetického a geografického pólu a magnetické inklinace. Ta se provádí tak, že se kompas podle referenčního přístroje nasměruje na sever, a přes sběrnici IIC se zahájí kalibrace. Na následujícím obrázku je vidět rozložení magnetické deklinace vzhledem k poloze na zemi v roce 2000. Rok se uvádí záměrně, protože se mapa relativně rychle mění, a měla by být měřena asi dvakrát ročně. Hodnota deklinace je pozitivní, když magnetický pól je východně od geografického. Takže například hodnota 10 znamená, že od směru na magnetický pól musíme odečíst 10° východně, abychom získali směr na geografický pól.



Obr. 7 Světová mapa magnetické deklinace (ve stupních východně)

Na obrázku kompasu (Obr. 6) jsou pojmenovány jednotlivé piny výstupního konektoru. Pin 1 a 9 slouží pro přivedení napájecího napětí (5V). Vodiče sběrnice IIC se připojují k pinům 2 a 3. Kompas poskytuje údaj o směru magnetického pole i prostřednictvím PWM výstupu. Ten je zakódován do šířky pulzu, který se pohybuje v rozmezí od 1 mS (pro 0°) do 36.99 mS (pro 359.9°). Jinými slovy je to 100 uS/° s 1ms ofsetem. Signál je mezi pulzy v úrovni logické nuly vždy po dobu 65 ms. Pokud se nevyužívá sběrnice IIC, mohou se pro kalibraci měření využít piny 5 a 6. Sestupnou hranou na pinu 6 se provede dílčí část kalibrace. Celkem se musí přivést čtyři takovéto sestupné hrany. Na pinu 5 můžeme sledovat stav kalibrace, ten je v jejím průběhu v úrovni logické nuly.

Standardní adresa kompasu na sběrnici IIC je 0x60. Komunikace se zahajuje zapsáním adresy následované číslem registru, se kterým chceme pracovat (číst/zapisovat). Poté po opětovném zaslání adresy můžeme rovnou číst, nebo zapisovat do požadovaného registru. Seznam dostupných registrů je v následující tabulce (Tab. 1).

Registr	Funkce
0	Číslo revize software
1	Azimut jako byte (0-255 pro celý kruh)
2, 3	Azimut jako word (0-3599 pro celý kruh)
4, 5	Interní test – Senzor 1
6, 7	Interní test – Senzor 2
8, 9	Interní test – Senzor 1 – surová data
10, 11	Interní test – Senzor 2 – surová data
12	Unlock code1 – pro změnu IIC adresy
13	Unlock code2
14	Unlock code3
15	Command register

Tab. 1 Jednotlivé registry pro ovládání kompasu přes sběrnici IIC

Z registrů nás nejvíce zajímá azimut ve formě 16bitové bezznaménkové hodnoty, tedy registry 2 a 3. Po přečtení hodnoty stačí pouze toto číslo vydělit desíti, a dostaneme přímo hodnotu natočení.

[3] [15]

GPS přijímač

Pro příjem GPS signálu je vybrán kompaktní modul firmy ORCAM System označený zkratkou GPS35. Jedná se o přijímač s architekturou SiFR Star III, 20 kanálový a s podporou komunikačních protokolů NMEA 0183 a SiFR Binary.



Obr. 8 GPS přijímač GPS35

Zařízení pracuje s 3V logikou, ale jelikož obsahuje integrovaný stabilizátor napětí, může se napájet napětím v rozsahu 3,3 – 5,5 V DC. Ke komunikaci se zařízením slouží dva sériové porty s napěťovou úrovní CMOS. Na portu A je standardně nastaven protokol NMEA 0183 s komunikační rychlostí 9600 nebo 4800 baud.

Přijímač pro svůj chod potřebuje externí aktivní anténu. Napájení antény se zajišťuje buďto z externího zdroje, anebo také ze zdroje integrovaného v přijímači, stačí pouze propojit dva piny na přijímači.



Obr. 9 Aktivní GPS anténa

Protokol NMEA definuje strukturu a obsah zpráv, které by měly podporovat všechny GPS přijímače. Přijímač GPS35 podporuje mimo jiné tyto typy výstupních zpráv:

Označení	Obsah
GGA	Zeměpisná délka a šířka, geodetická výška, čas určení souřadnic
GLL	Zeměpisná délka a šířka, UTC čas
GSA	Aktivní satelity a DOP
GSV	Informace o družicích
RMC	Minimální doporučená informace pro navigaci
VTG	Směr a rychlost vztahená k povrchu Země

Tab. 2 Výstupní NMEA zprávy podporované přijímačem GPS35

Nejvíce používaná zpráva je bezesporu zpráva s označením RMC (Recommended Minimum Specific GNSS Data), tedy doporučené minimum přenášených dat. Například:

\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, ,*10

Význam jednotlivých segmentů zprávy vysvětluje následující tabulka.

\$GPRMC	Hlavička označující zprávu RMC
161229.487	UTC čas, ve formátu: hhmmss.sss
A	Status, A = platné data, V = neplatné data
3723.2475	Zeměpisná šířka ve formátu: ddmm.mmmm
N	N/S indikace, N = sever, S = jih
12158.3416	Zeměpisná délka ve formátu: dddmm.mmmm
W	E/W indikace, W = západ, E = východ
0.13	Rychlost v uzlech (1uzel = 1853m/h)
309.62	Směr pohybu ve stupních
120598	Datum ve formátu: ddmmyy
	Nevyužívá se
*10	Kontrolní součet
<CR><LF>	Ukončovací znaky zprávy

Tab. 3 Význam věty RMC ve formátu NMEA

Kontrolní součet je vypočítáván tak, že mimo zahajovacího znaku „\$” a ukončovacího znaku „*“ se provede logická operace XOR, a to takto: $A = A \text{ xor } B$, kde A je výsledné CRC a B jsou znaky ve formátu ASCII.

[1] [12] [16]

Akcelerometr

Je vybrán tříosý akcelerometr MMA7261QT pracující na kapacitním principu. Výstupy jednotlivých os jsou analogové s integrovaným dolno-propustným filtrem prvního řádu s mezním kmitočtem pro osy XY na hodnotě 350 Hz a pro osu Z na hodnotě 150 Hz. Pomocí dvou digitálních vstupů lze vybírat mezi čtyřmi citlivostmi: 2.5g, 3.3g, 6.7g, 10g. Akcelerometr je proveden v pouzdře QFN.

Inkrementální čidla pro odometrii

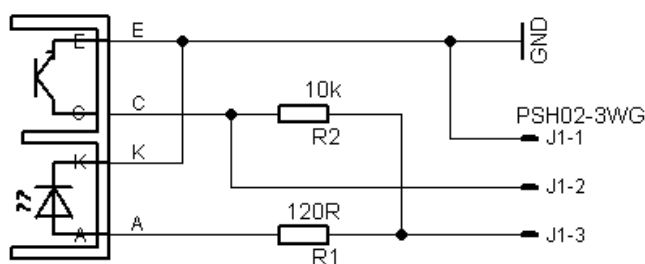
Pro měření ujeté dráhy je vybrán reflexní optočlen CNY70. Je to kompaktní součástka, která obsahuje jak zdroj světla ve formě infračervené diody s vlnovou délkou 950nm, tak i přijímací fototranzistor s optickým filtrem.

Na následujícím obrázku (Obr. 10) je senzor zobrazen i s deskou plošných spojů. Na ní jsou osazeny dva rezistory, jeden pro nastavení požadovaného proudu infračervenou LED diodou a druhý pro omezení maximálního proudu kolektorem fototranzistoru. A také konektor pro přivedení napájecího napětí a získání požadovaného signálu.



Obr. 10 Inkrementální čidlo

Následuje schéma zapojení (Obr. 11).



Obr. 11 Schéma zapojení podpůrné desky inkrementálního čidla CNY70

5.3 Snímače vzdálenosti

Ultrazvukový dálkoměr

Ultrazvukový senzor vzdálenosti je koupen jako hotový modul, se kterým se komunikuje prostřednictvím sběrnice IIC. Pracuje na principu měření doby od vyslání zvukového signálu po jeho příjem. Ultrazvukový signál je na kmitočtu 40kHz. Jeho rozsah je 3cm až 6m, s přesností 3 až 4cm. Dále je na modulu osazen senzor osvětlení, na jehož výstupu ale není číslo v žádných jednotkách, ale pouze hodnota od 0 do 255.



Obr. 12 Ultrazvukový senzor vzdálenosti SRF08

Standardní adresa senzoru na sběrnici IIC je 0xE0. Komunikace se zahajuje zapsáním adresy následované číslem registru, se kterým chceme pracovat (číst/zapisovat). Poté po opětovném zaslání adresy můžeme rovnou číst, nebo zapisovat do požadovaného registru. Seznam dostupných registrů je v následující tabulce (Tab. 4).

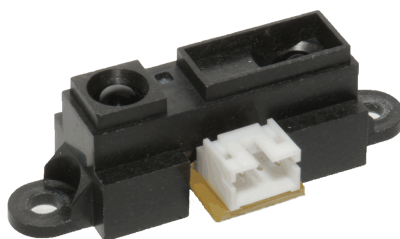
Registr	Funkce pro čtení	Funkce pro zápis
0	Verze software	Příkazový registr
1	Senzor osvětlení	Maximální zesílení
2	1 echo (horní byte)	Maximální dosah
3	1 echo (dolní byte)	nevyužito
~	~	~
34	17 echo (horní byte)	nevyužito
35	17 echo (dolní byte)	nevyužito

Tab. 4 Jednotlivé registry pro ovládání senzoru přes sběrnici IIC

Na nulté pozici se při zápisu nachází příkazový registr, kterým spustíme měření. Příkazem rovnou i určíme formát změřených dat. Můžeme si vybrat centimetry, palce nebo mikrosekundy. Výsledek se nám po 65ms objeví v registrech 2 až 35, což je sedmnáct 16bitových hodnot (rozdělených po bajtech). V prvním výsledku je vždy první zachycený odraz vyslaného signálu (nejbližší překážka). Ostatní mohou být nulové, nebo mohou obsahovat vzdálenosti k dalším, vzdálenějším objektům.

Infračervený dálkoměr

Infračerveným senzorem vzdálenosti je GP2Y0A21Y značky SHARP. Ten pracuje na principu polohově citlivého senzoru (PSD), a jeho měřicí rozsah je 10 až 80cm. Napájecí napětí je 5V, přičemž na výstupu je napětí úměrné vzdálenosti v rozsahu 0.4V až 3.2V.



Obr. 13 Infračervený senzor vzdálenosti GP2Y0A21Y

Laserový skener plochy SICK LMS100

Jako hlavní senzor pro snímání okolí byl vybrán laserový skener LMS100. Pracuje na principu měření doby od vyslání laserového impulzu po jeho zpětné zachycení po odrazu od překážky. Výsledkem jsou data ve dvojdimenzionálních polárních souřadnicích v rozsahu 270°/20m. Přičemž minimální re-emise měřeného povrchu, pro měření do vzdálenosti 20m, musí být 13%, což je přibližně bílá stěna v budovách. Pro kratší vzdálenosti (do 10 metrů) tato hodnota klesá až na 3%.

Mezi jeho hlavní rysy patří:

- Maximální dosah (poloměr plochy) - 20m
- Plošný skenovací úhel 270°
- Rozlišení mezi paprsky v ploše - 0,5°/0,25°
- Doba odezvy - 20ms
- Spínací vstupy/výstupy - 2/3
- Datová rozhraní - Ethernet, RS232, CAN
- Rozměry - 102mm x 152mm x 105mm
- Váha a krytí - 1,1kg; IP65
- Napájení 9 – 30 V DC



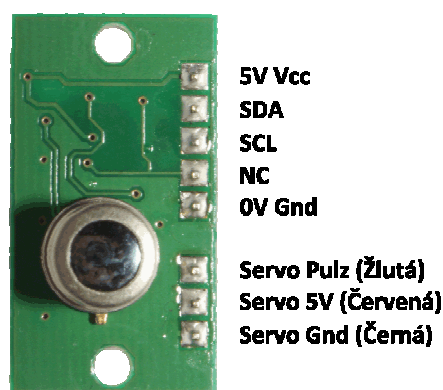
Obr. 14 Laserový dálkoměr SICK LMS100

Komunikace je možná přes několik různých rozhraní, ale ne přes všechna lze dosáhnout takových rychlostí, které by dostačovaly pro přenos „surových“ dat z čidla, tzn. všech 1080 vzorků vzdálenosti, každých 20ms. K tomu je určeno komunikační rozhraní ethernet. Z tohoto důvodu je senzor připojen přímo k hlavnímu řídicímu modulu, a dál se jím v této práci nebudeme zabývat.

5.4 Snímače okolního prostředí

Termočláňkové pole

Termočláňkové pole TPA 81 detekuje infračervené záření v rozsahu 2 až 22 μm , což je vlastně vlnová délka tepelného záření. Toto čidlo obsahuje osm termočláňků uspořádaných do sloupce. To umožňuje měření teploty osmi míst zároveň. Na čidle je umístěn PWM výstup pro řízení servo pohonu, s jehož pomocí je možné vytvořit tepelný obraz prostoru před vozidlem. Tímto čidlem lze rozpoznat plamen svíčky vzdálené dva metry od čidla. Funkce termočláňků není ovlivněná okolním světlem.



Obr. 15 Popis pinů čidla TPA 81

Typické zorné pole TPA 81 je $41^\circ \times 6^\circ$, což znamená, že na jeden termočlánek (pixel) připadá $5,12^\circ \times 6^\circ$.

Čidlo je osazeno pěti pinovým konektorem, kterým se přivádí napájení a je zde vyvedena sběrnice IIC (adresa 0x68) přes které se čidlo ovládá. Význam jednotlivých pinů čidla TPA 81 je popsán na předcházejícím obrázku (Obr. 15). Pin “Not Connect” by měl zůstat nezapojený, protože se jedná se o MCLR pin a je využit jen při programování procesoru (PIC16F88) po osazení DPS čidla. Servo pohon se připojuje přímo k pinům pro to určených. Napájení pro servo se bere z napájecího napětí čidla (5 V).

Ovládání čidla se uskutečňuje pomocí deseti registrů. Popis jednotlivých registrů je uveden v následující tabulce (Tab. 5).

Registr	Funkce při čtení	Funkce při zápisu
0	Revize softwaru	Příkazový registr (Command register)
1	Teplota okolí [$^\circ\text{C}$]	Rozsah polohy serva (verze 6 nebo vyšší)
2	Teplota pro pixel 1 [$^\circ\text{C}$]	N/A
3	Teplota pro pixel 2 [$^\circ\text{C}$]	N/A
4	Teplota pro pixel 3 [$^\circ\text{C}$]	N/A
5	Teplota pro pixel 4 [$^\circ\text{C}$]	N/A
6	Teplota pro pixel 5 [$^\circ\text{C}$]	N/A
7	Teplota pro pixel 6 [$^\circ\text{C}$]	N/A
8	Teplota pro pixel 7 [$^\circ\text{C}$]	N/A
9	Teplota pro pixel 8 [$^\circ\text{C}$]	N/A

Tab. 5 Seznam dostupných registrů v TPA81

Pouze do nultého a prvního registru lze zapisovat. Registr 0 je příkazový registr a využívá se pro nastavení pozice servo pohonu, a také pro změnu adresy zařízení na sběrnici IIC. Čtením tohoto registru získáme číslo revize softwaru. Zápisem do registru 1 lze nastavit rozsah otáčení servo pohonu. Přečtením tohoto registru dostaneme hodnotu teploty okolí v $^\circ\text{C}$, která se měří uvnitř senzoru. Ostatní registry lze pouze číst a představují jednotlivé teploty pixelů v $^\circ\text{C}$.

Pro správné odečítání teplot jednotlivých pixelů je důležité v každé pozici setrvat alespoň 40ms před přečtením hodnot z čidla.

Možné příkazy v příkazovém registru (Command register) jsou uvedeny v Tab. 6. Příkazy 0 až 30 nastavují hodnotu otočení servo pohonu. K dispozici máme tedy 31 kroků, které reprezentují posun o 180°. Poloha 15 je střenění poloha.

Příkaz	Popis
0	Minimální pozice serva
⋮	⋮
30	Maximální pozice serva
160	První sekvence změny adresy
165	Třetí sekvence změny adresy
170	Druhá sekvence změny adresy

Tab. 6 Seznam příkazů příkazového registru

Čidlo generuje PWM signál, jehož šířku impulzu lze vypočítat pomocí následujícího vzorce:

$$t_H = \text{Pozice Serva} \times 60 + 540 \quad [\mu\text{s}; -] \quad (1)$$

Z čehož vyplývá, že minimální délka pulzu je 540 μs pro pozice serva rovnou 0 a maximální délka pulzu je 2340 μs pro pozici serva rovnou 30. V tomto případě je minimální změna velikosti pulzu 60 μs. Zapsání jakékoliv jiné hodnoty než 0 až 30, způsobí nastavení serva do prostřední pozice ($t_H = 1500 \mu\text{s}$).

Ve verzi software vyšším než 6 je možné zápisem do registru 1 nastavit minimální změnu šířky impulzu PWM signálu (krok). Standardní krok je již předem zmíněných 60 μs. Zápisem hodnoty 20 až 120 je možné nastavit požadovanou velikost kroku, a tím i úhel otáčení. Výpočet kroku, lze povést následujícím vzorcem.

$$t_\Delta = \frac{\text{Rozsah serva}}{2} \quad [\mu\text{s}; -] \quad (2)$$

Pro hodnotu 20 odpovídá minimální krok 10 μs a hodnotě 120 odpovídá krok 60 μs. Například u velikosti kroku 10 μs může řídicí impuls nabývat šířky 1345 μs až 1655 μs a pro velikost kroku 60 μs může řídicí impuls nabývat šířky 570 μs až 2430 μs.

[2]

Tlak

Pro měření okolního tlaku vzduchu je vybrán senzor MPX4115A od výrobce Freescale. Ten měří hodnotu absolutního tlaku v rozsahu od 15 do 115kPa. Obvod vyžaduje napájení 5V a na jeho výstupu je napětí úměrné tlaku.

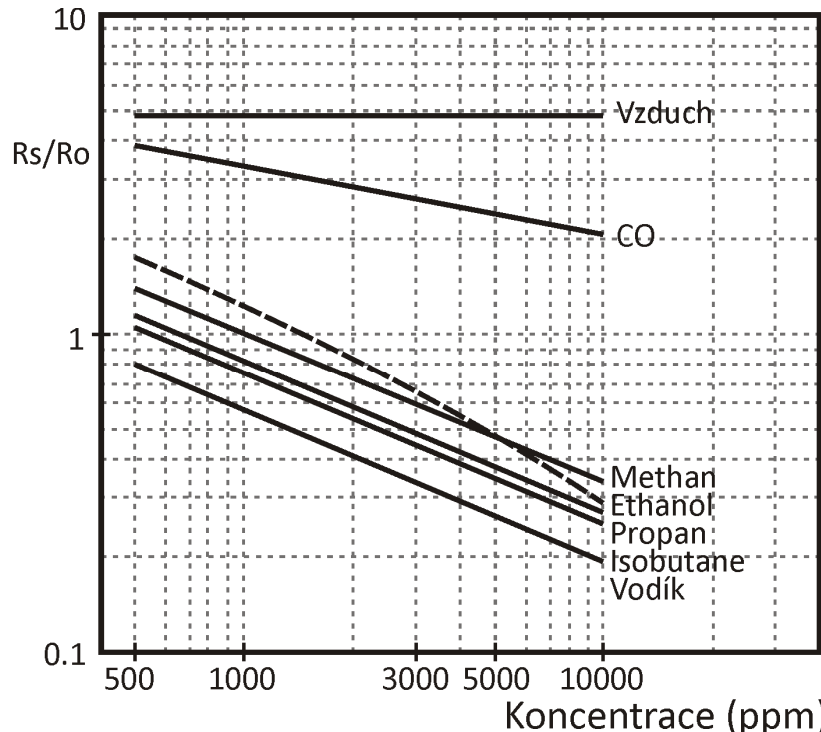
Výstupní napětí je určeno následujícím vzorcem:

$$V_{OUT} = V_S \cdot (0.009 \cdot P - 0.095) \pm \text{Chyba} \quad [V; V, kPa, kPa] \quad (3)$$

Kde V_{OUT} je výstupní napětí senzoru, V_S je napájecí napětí, P je měřený absolutní tlak a „Chyba“ je hodnota v kPa, o kterou se měřená hodnota může lišit od skutečné. Ta je určena grafy výrobce a pro rozmezí teplot 0° až 85°C je rovna 1.5kPa.

Senzor hořlavých plynů

Pro měření složení okolního plynu je vybrán senzor hořlavých plynů TGS813 firmy FIGARO. Senzor reaguje změnou svého vnitřního odporu na několik různých hořlavých plynů. To sice znemožňuje určení, o jaký plyn se přesně jedná, ale pro účely vozidla, které se nesnaží určit přesné složení okolních plynů, ale jen přítomnost plynů hořlavých, to dostačuje. Na následujícím grafu závislosti (Obr. 16) vidíme citlivost na jednotlivé plyny. Závislá osa má význam poměru odporů R_s ku R_o , kde R_s je proměnný odpor senzoru a R_o je odpor senzoru při koncentraci metanu o hodnotě 1000ppm.



Obr. 16 Závislost poměru odporů R_s/R_o ke koncentraci hořlavých plynů

6 Návrh modulů

6.1 Výběr vhodné platformy

Při výběru platformy byl kladen důraz na dostupnost vývojových prostředků, přítomnost standardních rozhraní jako ADC, sběrnice IIC, SCI, SPI a dostatečný výpočetní výkon.

Pro řízení modulů je v obou případech využito mikroprocesoru Freescale mc9s12dp512. Ten v sobě zahrnuje velké množství periférií pro komunikaci a sběr dat z okolí. Procesor je hojně využíván v automobilovém průmyslu, a zároveň se s ním pracuje i v laboratořích školy, a tím pádem jsou k němu běžně dostupné i vývojové nástroje.

mc9s12dp512 je 16bitový mikroprocesor skládající se ze standardních integrovaných periférií, které jsou uvedeny v následujícím přehledu:

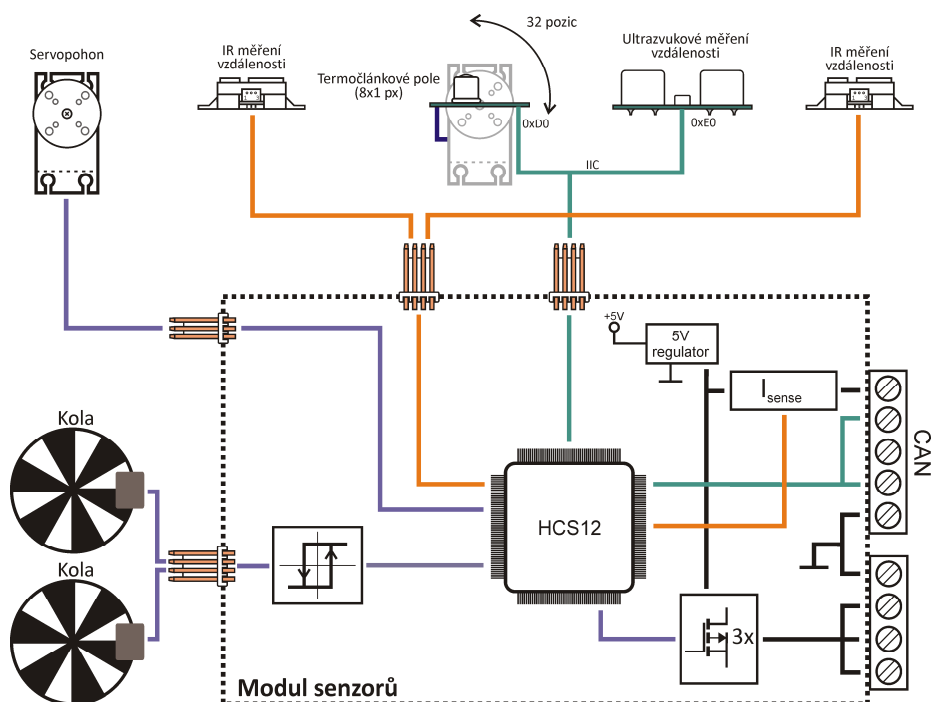
- 16bitová šířka sběrnice
- Rychlost sběrnice až 25 MHz
- 512 kB Flash EEPROM paměti pro program
- 14 kB RAM paměti pro data
- 4 kB EEPROM pro data
- 2x sériové komunikační rozhraní SCI
- 3x sériové periferní rozhraní SPI
- Osmikanálový rozšířený záchytný systém / časovač
- 2x osmikanálový 10bitový AD převodník (ADC)
- Osmikanálová pulzně šířková modulace (PWM)
- Sběrnice IIC
- 5x modul CAN 2.0 A, B s maximální přenosovou rychlostí 1 Mbps

[8]

6.2 Návrh modulu snímačů

Na vozidle jsou poříty dva moduly snímačů. Jeden na přední části a druhý na zádi. V obou případech k němu jsou připojeny dvě infračervené a jedno ultrazvukové čidlo vzdálenosti, a dvě reflexní závory pro snímání ujeté dráhy. K modulu umístěnému v přední části je navíc připojeno servo pro ovládání natočení předních kol a otočené termočlávkové pole.

Podrobnější blokové schéma je zobrazeno na následujícím obrázku (Obr. 17). Infračervené snímače vzdálenosti mají napěťové výstupy, a jsou proto připojeny k AD převodníku procesoru. Ultrazvukový snímač vzdálenosti a termočlánekové pole je ovládáno prostřednictvím sběrnice IIC. Signál z reflexních závor vede od kol přes obvod hystereze na čítačový vstup procesoru. Servo pohon pro natáčení kol je ovládán jedním PWM kanálem. Na modulu se nacházejí i tři tranzistory pro spínání libovolné zátěže (například světel), a spolu se sběrnici CAN jsou vyvedeny na 9pinový konektor. Celkový proud spotřebovávaný modulem je měřen a vyhodnocován.

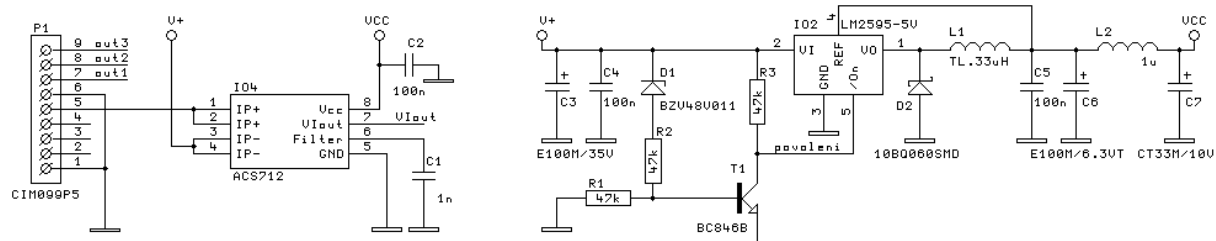


Obr. 17 Blokové schéma modulu snímačů a jeho periferních zařízení

Schéma zapojení i deska plošných spojů byly navrženy s využitím programu Formica 4.40. V následující části textu budou popsány důležité části zapojení. Celkové schéma zapojení modulu snímačů naleznete v přílohách.

6.2.1 Spínaný zdroj s měřením proudu

Požadavkem bylo zajistit napájení mikroprocesoru a jeho periférií napětím 5 V s maximálním odebíraným proudem 1 A, a zároveň měření odebíraného proudu jak samotným mikroprocesorem, tak i zátěží spínanou výkonovými tranzistory. Na následujícím obrázku (Obr. 18) je zobrazeno schéma zapojení.



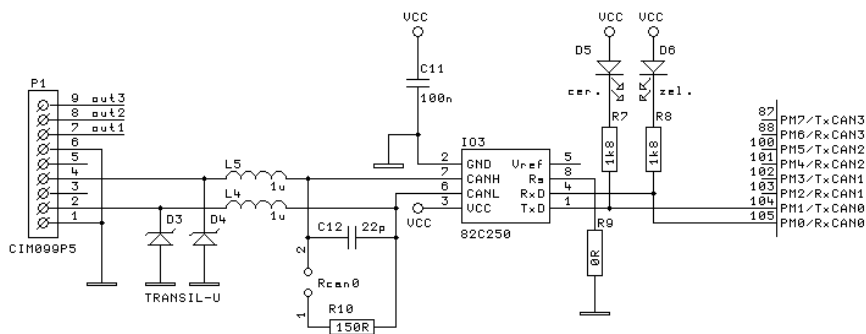
Obr. 18 Schéma zapojení spínaného zdroje a měření proudu

Napájení je do zařízení přivedeno přes 9pinový konektor P1. Integrovaný obvod ACS712 (IO4) pracující na Halově principu měří procházející proud, přičemž na jeho výstupu je napětí úměrné tomuto proudu. Kondenzátor C1 slouží k nastavení mezního kmitočtu dolní propusti umístěné na výstupu tohoto obvodu. Pro hodnotu 1 nF to znamená mezní kmitočet 80 kHz.

Jako stabilizátor napětí je vybrán obvod LM2595-5V (IO2), kvůli jeho dobré dostupnosti a dobrým vlastnostem. Jeho obvodové zapojení vychází z elektronických vývojových desek provozovaných v laboratořích školy. Rezistory R1, R2, R3, dioda D1 a bipolární tranzistor T1 zajišťují zapnutí stabilizátoru až od hodnoty vstupního napětí přibližně od hodnoty 5.4 V. Dle požadovaného výstupního napětí, výstupního proudu a maximálního vstupního napětí, výrobce udává, že se teplota na povrchu součástky (bez použití chladiče) zvýší maximálně o 45°C nad úroveň okolí.

6.2.2 Zapojení sběrnice CAN

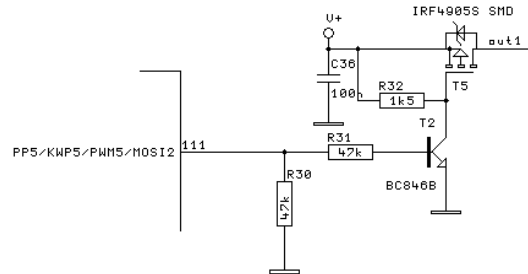
Zapojení obvodů sběrnice CAN opět vychází ze zapojení elektronických vývojových desek (EVBHCS12) provozovaných v laboratořích školy. Jako řadič fyzické vrstvy slouží obvod 82C250 (IO3). Prostřednictvím zkratovací propojky „Rcan0“ lze do obvodu zařadit terminační odpor sběrnice R10. Transily D3 a D4 na vodičích sběrnice slouží jako ochrana proti přepětí, a luminiscenční diody D5 a D6 na ni indikují provoz.



Obr. 19 Schéma zapojení sběrnice CAN

6.2.3 Zapojení unipolárních tranzistorů

Požadavkem bylo zajistit spínání třech zátěží s maximálním procházejícím proudem 5 A a napětím 12 V. Na následujícím obrázku (Obr. 20) je vyobrazeno schéma zapojení. Toto zapojení se v celkovém schématu vyskytuje třikrát.



Obr. 20 Schéma zapojení unipolárních tranzistorů

Jako spínací prvek je vybrán unipolární tranzistor (T5) s kanálem typu P (IRF4905S). Jeho maximální proud I_{DS} je -74A a odpor kanálu v sepnutém stavu je pouhých $R_{DS(on)} = 0.02\Omega$. Jeho maximální výkonová ztráta bude:

$$P_{T5} = R_{DS(on)} I_{DS}^2 = 0.02 \cdot 5^2 = \underline{0.5 W} \quad (4)$$

Teplota jeho přechodu při okolní teplotě 25 °C bude:

$$v_j = P_{T5} \cdot R_{th(j-a)} + v_a = 0.5 \cdot 40 + 25 = \underline{45 ^\circ C} \quad (5)$$

Rezistor R30 zajišťuje uzavření bipolárního tranzistoru T2 (odolnost proti rušení) a rezistor R31 omezuje proud tekoucí do jeho báze na:

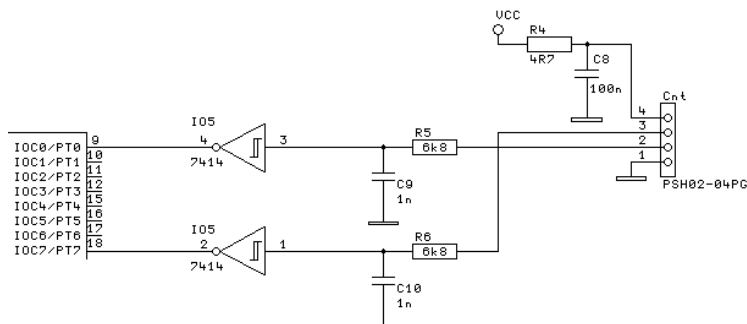
$$I_B = \frac{U_H - U_{BE}}{R31} = \frac{5 - 0.6}{47000} = \underline{94 \mu A} \quad (6)$$

Rezistor R32 přivádí napětí pro uzavření unipolárního tranzistoru T5 při rozepnutém T2, a také omezuje kolektorový proud tranzistoru T2 na:

$$I_C = \frac{V}{R32} = \frac{12}{1500} = \underline{8 mA} \quad (7)$$

6.2.4 Připojení inkrementálních čidel

Reflexní optozávory se k modulu připojují přes 4pinový konektor, na kterém je vyvedeno i napájení. Signál ze senzorů putuje přes dolno-propustný filtr nastavený na frekvenci 23.5kHz, Schmidtův klopný obvod (IO5) a vstupuje do 16bitového pulzního akumulátoru mikroprocesoru.

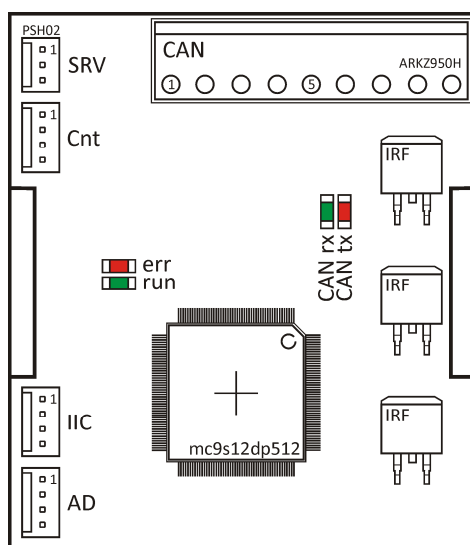


Obr. 21 Schéma zapojení obvodu inkrementálních čidel

6.2.5 Rozložení desky plošných spojů

Deska plošných spojů je navržena na rozměry 67 x 77mm pro umístění do krabice WEB1002. Tato krabice je pro vývoj výhodná, protože jsou součástky dobře přístupné a vyrábí se jako stavebnice, ze které lze sestavit potřebnou velikost. Na následujícím obrázku (Obr. 22) je zobrazeno rozložení důležitých součástí modulu. Při horním okraji je umístěn 9pinový konektor sběrnice CAN a výstupu výkonových tranzistorů. Na levé straně jsou shora konektory pro připojení servo pohonu, reflexních optozávor, sběrnice IIC a pro připojení infračervených dálkoměrů. Na desce se nacházejí dva páry luminiscenčních diod. Diody s označením „err“ a „run“ slouží k indikaci stavu, ve kterém se modul, popřípadě operační systém, nachází. Druhý pár jsou již výše zmíněné signalizace provozu na sběrnici CAN. Výkonové tranzistory jsou umístěny na pravé straně plošného spoje. První shora spíná třetí výstup, prostředí spíná druhý a poslední spíná výstup první.

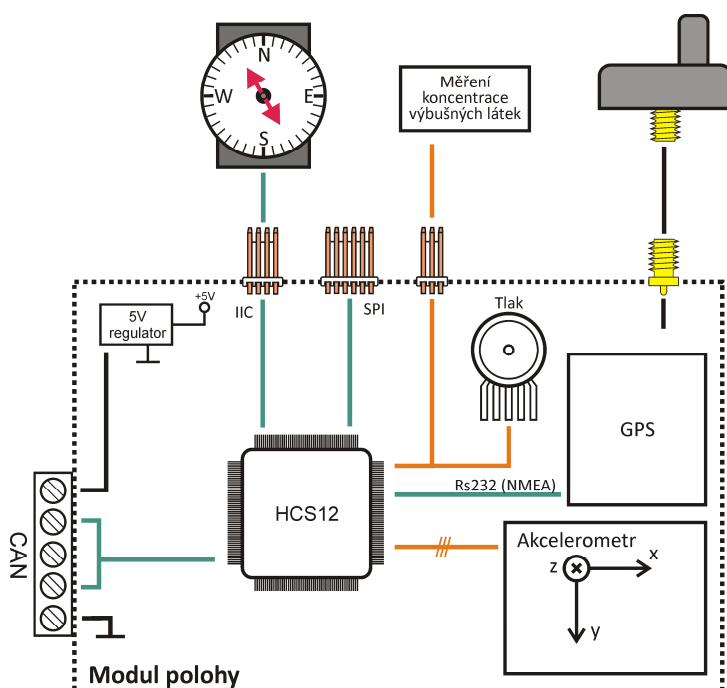
Fotodokumentace vyrobeného a osazeného modulu senzorů se nachází v přílohách.



Obr. 22 Rozložení desky plošných spojů modulu senzorů

6.3 Návrh modulu polohy

Na vozidle je umístěn jeden modul polohy, který shromažďuje a udržuje informace o jeho poloze a natočení, a samozřejmě také informace o jeho okolním prostředí. Podrobnější blokové schéma je zobrazeno na následujícím obrázku (Obr. 23). Zařízení je opět napájeno ze sběrnice CAN prostřednictvím spínaného zdroje napětí. Přímo na desce plošných spojů je umístěn akcelerometr pro určování náklonu vozidla, GPS přijímač s konektorem pro aktivní anténu, a také čidlo absolutního tlaku. Prostřednictvím sběrnice IIC je připojen elektronický kompas. Je tomu tak z důvodu, aby ho bylo možné umístit dále od zdrojů rušivých signálů, jako jsou spínané zdroje anebo motor pro pohon vozidla. Dále od desky plošných spojů je také vyveden senzor pro měření koncentrace výbušných látek v ovzduší. Je to proto, že se senzor při své činnosti zahřívá, a mohl by teplotně ovlivňovat ostatní části modulu.



Obr. 23 Blokové schéma modulu polohy a jeho periferních zařízení

V následující části textu budou popsány důležité části zapojení. Celkové schéma zapojení modulu polohy naleznete v přílohách. Některé části, jako například zapojení spínaného zdroje nebo sběrnice CAN, se shodují s předchozím modulem senzorů, a proto se jimi již nebudeme zabývat.

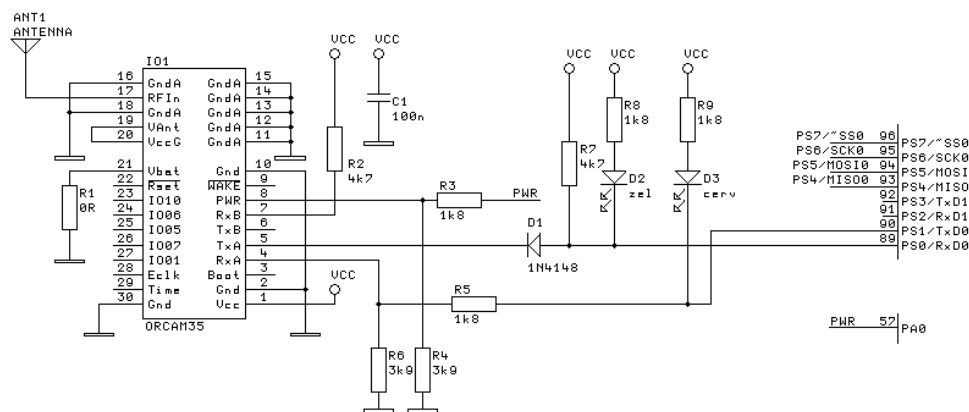
6.3.1 Zapojení GPS přijímače

GPS přijímač ORCAM35 je uzpůsoben pro nasazení v mobilních aplikacích, kde se často napájí přímo z baterie, a proto má široký rozsah napájení. I když jsou jeho vstupy/výstupy typu CMOS, nebylo nutné pro něj napájecí napětí dále snižovat. (Obr. 24)

Jako rozhraní mezi mikroprocesorem a přijímačem slouží asynchronní sériová linka, která ale pracuje na různých napěťových úrovních. Z toho důvodu byl vytvořen napěťový dělič odpory R5 a R6, který sníží výstupní signál z procesoru přibližně na dvě třetiny. Opačným směrem je nutné

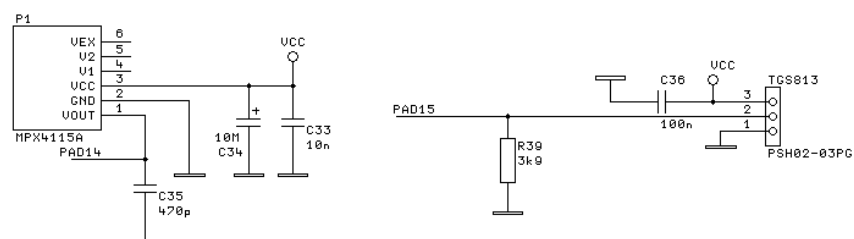
výstupní napětí z GPS přijímače zvýšit na úroveň TTL logiky mikroprocesoru. To je provedeno zapojením diody D1 a odporu R7. Na diodě vzniká úbytek napětí, který se přičítá k výstupnímu napětí z GPS přijímače. Signály sériové linky jsou opatřeny luminiscenčními diodami D2 a D3 pro signalizaci provozu.

Samotný GPS přijímač nepotřebuje téměř žádné další externí součástky. Výběr pracovního režimu se provádí pinem PWR (8), který je napojen na mikroprocesor opět přes napěťový dělič. Napájení aktivní antény poskytuje přímo GPS přijímač, stačí pouze propojit piny „VccG“ (20) a „VAnt“ (19). Vstupní pin „Vbat“ (21) slouží pro připojení záložní baterie, tu ale nevyužíváme, a proto je pin připojen k napájecí zemi. Odpor R2 slouží k definování klidové úrovně při nezapojení vstupu druhé sériové linky.



6.3.3 Senzor tlaku a senzor koncentrace výbušného plynu

Pro měření absolutního tlaku se využívá senzor MPX4115A firmy Freescale. Ten vyžaduje pouze připojení napájecího napětí, ze kterého generuje analogový signál úměrný okolnímu tlaku. Tento signál je měřen analogově digitálním převodníkem procesoru. Podobně nenáročný je i senzor koncentrace výbušných látek TGS813 firmy FIGARO. Postačuje mu napájení a na analogovém výstupu zařazený odpor R39, na kterém se vytváří napětí.

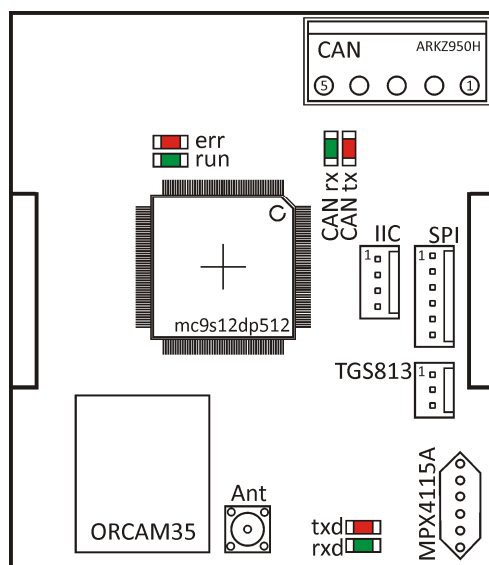


Obr. 26 Zapojení tlakového čidla MPX4115A (vlevo) a senzoru výbušných plynů TGS813

6.3.4 Rozložení desky plošných spojů

Deska plošných spojů je navržena na rozměry 67 x 77mm pro umístění do krabice WEB1002. Na následujícím obrázku (Obr. 27) je zobrazeno rozložení důležitých součástí modulu. V horním pravém rohu se nachází konektor pro napájení a sběrnici CAN. Těsně pod ním jsou luminiscenční diody pro signalizace provozu na sběrnici CAN. Na pravé straně se nachází konektory pro připojení elektronického kompasu sběrnicí IIC, vyvedená sběrnice SPI a konektor pro připojení senzoru výbušných látek. V pravém dolním rohu je umístěn tlakový senzor. GPS přijímač s konektorem RSMA a signalizačními diodami se nachází ve spodní části plošného spoje.

Fotodokumentace vyrobeného a osazeného modulu polohy se nachází v přílohách.



Obr. 27 Rozložení desky plošných spojů modulu polohy

7 Operační systém

Existují dvě řešení jak navrhnout softwarovou strukturu požadovaného zařízení. Tedy s operačním systémem, nebo bez něj. Implementace zařízení bez operačního systému má své výhody, jako například cena, nebo menší náročnost na výpočetní výkon, ale u složitějších úloh jako je tato, převažují jeho nevýhody. To může být například nízká variabilita a špatná přenositelnost kódu.

Často navíc sama řízená úloha přímo vybízí k použití více-vláknového zpracování, což je jasný signál pro použití operačního systému. Ulehčíme si tak práci využitím předem připravených synchronizačních nástrojů pro synchronizaci úloh a hardwarových přerušení.

Shrme-li předchozí, využití operačního systému dává smysl u složitějších úloh, kde předpokládáme častou modifikaci programu (například přidání dalších senzorů), a proto je u všech modulů na robotickém zařízení využít operační systém.

Jednotlivé řešení a jeho výhody porovnáme v následující tabulce (Tab. 7).

Bez operačního systému	S operačním systémem
+ jednoduchost	+ variabilita
+ cena vývoje	+ přenositelnost kódu
- přenositelnost kódu	+ synchronizační nástroje
	- větší nároky na výpočetní výkon
	- větší nároky na velikost paměti

Tab. 7 Porovnání struktur software

7.1 Výběr vhodného operačního systému

Existuje celá řada dobrých operačních systémů (Tab. 8) určených, nebo alespoň částečně určených, pro malé vestavné (embedded) systémy, postavených na méně výkonných mikroprocesorech.

Po zvážení všech kladů a záporů operačních systémů různých výrobců, byl vybrán výrobce FreeRTOS a jeho stejnojmenný operační systém reálného času. Mezi jeho hlavní výhody patří:

- Dostupné demonstrační aplikace pro různé architektury
- Je šířen pod upravenou GNU licenci (nemusí se publikovat zdrojové kódy)
- Je zdarma, volně stažitelný
- Má nízké nároky na výpočetní výkon, dá se aplikovat i na 8bitové mikroprocesory
- Podporuje až 23 různých architektur

Výrobce	Název operačního systému
AVIX	AVIX-RT 2.2.1
CMX Systems	CMX 5.30
Express Logic	ThreadX G5.1.5.0
FreeRTOS	FreeRTOS v6.0.2
Micrium	μC/OS-II v2.84
Pumpkin	Salvo 4 Pro and Salvo 4 LE
RoweBots	DSPNanoUnison
Segger	embOS V3.52
OSEK	OSEK/VDX

Tab. 8 Operační systémy různých výrobců

7.2 FreeRTOS

FreeRTOS je operační systém reálného času pro vestavné zařízení. Téměř celý operační systém je napsán v programovacím jazyce C, pouze několik funkcí je vytvořeno v assembleru. Podporuje jak plně preemptivní tak kooperativní zpracování vláken. Jádro tohoto RTOS je složeno pouze ze tří souborů, a to *task.c*, *queue.c* a *list.c*.

FreeRTOS je jednou ze tří distribucí, které výrobce poskytuje. Zbývající dvě verze jsou:

- **OpenRTOS** – Komerční verze FreeRTOS, která sebou nese plnou podporu USB a TCP/IP stacku
- **SafeRTOS** – Certifikovaná verze pro bezpečnostní vestavné aplikace. Splňuje normu SIL3 dle IEC61508

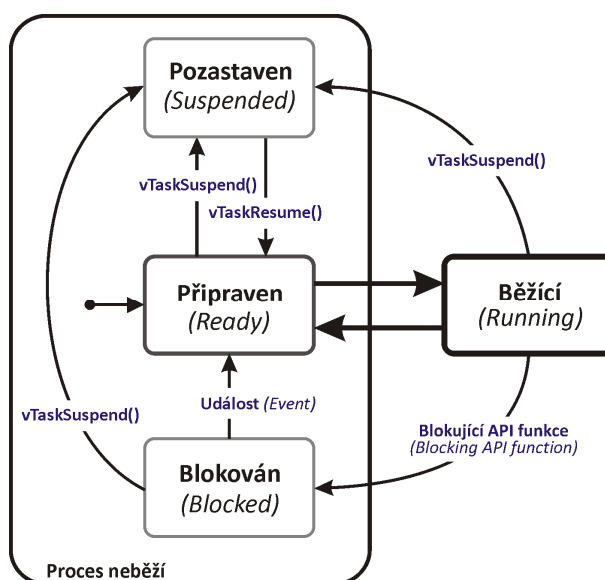
Pro komunikaci mezi vlákny, nebo mezi přerušeními a vlákny poskytuje komunikační a synchronizační nástroje typu *fronta zpráv*, (*binární*) *semafor* a *mutex*. Nástroj mutex zároveň řeší inverzi priorit.

7.2.1 Práce s vlákny – plánovač

Srdcem každého operačního systému je plánovač (scheduler). FreeRTOS disponuje plánovačem s prioritním plánováním, kdy je každému vláknu přiřazena hodnota jeho priority. Menší číslo znamená nižší přiřazenou prioritu, přičemž nejnižší možná hodnota je 0. V případě vláken se stejnou prioritou jsou vlákna plánována metodou „kruhu“ (Round Robin). To znamená, že jsou pravidelně střídána. Maximální priorita je omezena na námi požadovanou hodnotu, kterou lze nastavit v souboru „FreeRTOSConfig.h“

Na následujícím obrázku (Obr. 28) jsou znázorněny stavy, ve kterých se vlákno může nacházet. V jednom okamžiku může být procesoru přiděleno pouze jedno, a to odpovídá stavu „Běžící“. Ostatní vlákna jsou v tom okamžiku ve stavu „neběží“, který se pak dále rozděluje:

- **Pozastaven** (Suspended) – Vlákna v tomto stavu nejsou přístupná plánovači. Každé vlákno lze umístit a vyjmout z tohoto stavu funkcemi *vTaskSuspend()* a *vTaskResume()*
- **Připraven** (Ready) – Vlákna v tomto stavu jsou připravena ke zpracování a čekají na přidělení procesorového času plánovačem
- **Blokován** (Blocked) – Vlákna čekající na událost. Události mohou být časové (vlákno je zablokováno na určitý čas) a synchronizační (semafor, mutex, fronta zpráv)



Obr. 28 Stavy, ve kterých se vlákno může nacházet

[4] [5]

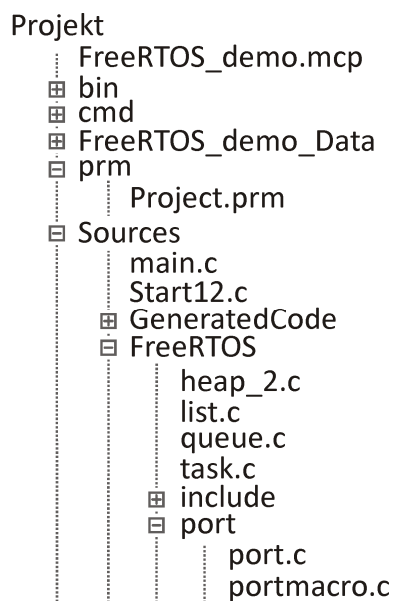
7.3 Úprava FreeRTOS pro procesor mc9s12dp512

I když výrobce poskytuje mnoho předpřipravených projektů pro různé druhy mikroprocesorů, není v jeho silách obsáhnout všechny typy různých výrobců. Pro náš požadovaný mikroprocesor výrobce takzvaný port (projekt pro dané zařízení) neposkytuje, a proto bylo nutné ho vytvořit.

Nejprve bylo experimentováno s úpravou již existujícího projektu pro mikroprocesor mc9s12dp256B, na kterém jsme si ověřili, jak vlastně tento operační systém funguje, a co musíme zajistit pro jeho chod. Poté byl vytvořen nový projekt v prostředí CodeWarrior verze 5.9.0 a do něj přidány soubory nutné k běhu systému.

7.3.1 Vytvoření nového projektu

Po vytvoření nového projektu v prostředí CodeWarrior pro požadovaný typ procesoru, bylo nutné přidat soubory operačního systému a vytvořit jeho stromovou strukturu. Výrobce poskytuje operační systém jako celek, ve kterém se nacházejí jak jeho zdrojové kódy, tak všechny dostupné „porty“ pro různé typy mikroprocesorů. Tento nepřehledný balík dat se ale nehodí pro vyvíjení aplikace, a proto byly vyjmuty pouze nutné soubory a vytvořena struktura, která je znázorněna na následujícím obrázku (Obr. 29).



Obr. 29 Struktura projektu s operačním systémem

Popis důležitých položek ve struktuře je následující:

- **Project.prm** – Mapování paměti procesoru
- **Sources** – Složka, ve které jsou uloženy zdrojové kódy
- **main.c** – V tomto souboru se nachází hlavní část uživatelského programu jako inicializace periferních zařízení, vytváření vláken a globálních synchronizačních nástrojů
- **GeneratedCode** – Tato složka obsahuje soubory, které definují takzvané „low level“ funkce. Ty se provádí při spouštění procesoru
- **FreeRTOS** – Tato složka obsahuje soubory operačního systému
- **list.c, queue.c, task.c** – Soubory tvořící jádro operačního systému
- **port.c, portmacro.c** – Tyto dva soubory tvoří takzvaný „Port“ operačního systému. Jsou to funkce, které jsou různé pro každou architekturu, a jsou potřeba pro chod systému

7.3.2 Generace časových přerušení pro plánovač

Ke generování časových přerušení pro plánování vláken (scheduler) je využito sedmého kanálu mikroprocesorové periférie ECT (Enhanced Capture Timer), neboli časovače.

Frekvence sběrnice mikroprocesoru je nastavena na 25 MHz, kterou je zároveň i inkrementován 16bitový čítač/časovač, jehož již zmíněný sedmý kanál generuje přerušení. Z toho plynou i omezení pro rozsah nastavení možných frekvencí spouštění plánovače.

Frekvence spouštění plánovače se nastavuje v konfiguračním souboru „FreeRTOSConfig.h“ položkou „configTICK_RATE_HZ“. Zde se může frekvence nastavit v rozsahu 382 Hz až 10000 Hz.

Výpočet minimální hodnoty (382 Hz):

$$f_{MIN} = \frac{f_{BUS}}{2^{16}} = \frac{25 \times 10^6}{2^{16}} = 381,469 \text{ Hz} \Rightarrow \underline{382 \text{ Hz}} \quad (8)$$

Maximální hodnota není omezena schopností časovače dosáhnout takového kmitočtu, ale časem potřebným k vykonání obsluhy přerušení, tedy přepnutí kontextu. Při zvolené hodnotě 10000 Hz se tedy bude přerušení vyvolávat každých 100 μ s. Měřením bylo zjištěno, že přepnutí kontextu (pouze dvě vlákna aktivní) trvá 8 μ s. Z toho vyplývá, že pro jednoduché aplikace je možné tento kmitočet teoreticky využít, ale pro složitější už se budeme blížit periodě přerušení a na samotnou aplikaci nezbude skoro žádný výpočetní čas.

Ke spuštění našeho časovače dochází v okamžiku volání funkce „vTaskStartScheduler()“. Jak má operační systém časovač nastavit mu definujeme v souboru port.c funkcí „prvSetupTimerInterrupt()“. Výpis této funkce je zobrazen v následující tabulce (Tab. 9). První tři instrukce slouží k nastavení módu časovače a jeho vstupní děličky kmitočtu. Následuje výpočet hodnoty *TempTC7*, která určuje periodu přerušení a je v jeho obsluze i využívána. Nakonec se časovač nastaví pro první spuštění a vynuluje se jeho příznak přerušení.

```
static void prvSetupTimerInterrupt( void )
{
    TIE = TIE | 128;           //TC7 povolení přerušení
    TSCR2 = TSCR2 | 0;        //Prescaler = 0
    TIOS = TIOS | 128;        //Kanál TC7 nastavit do módu Output Compare

    TempTC7 = 25000000/configTICK_RATE_HZ; //Výpočet požadované periody
                                           //přerušení časovače

    TC7 = TempTC7;            //Nastavení této periody

    TFLG1 = 128;              //Reset příznaku přerušení pro kanál TC7
}
```

Tab. 9 Výpis programu – funkce nastavující časovač pro plánování

Obsluha přerušení našeho časovače se nachází v souboru port.c pod názvem „vPortTickInterrupt()“. V ní dochází k přeplánování vláken. Jelikož se jedná o přerušení, musí být umístěna mimo stránkovanou oblast paměti programu. To je zajištěno instrukcí preprocesoru „#pragma CODE_SEG __NEAR_SEG NON_BANKED“ před samotnou funkcí. Jako první se ve funkci vypočítává a nastavuje počet inkrementů časovače pro další přerušení. Za ní následuje rozhodovací instrukce preprocesoru, která určuje typ nastaveného plánování (preemptivní nebo kooperativní multitasking). Při preemptivní typu plánování dojde k uložení kontextu právě prováděného vlákna, k inkrementaci počítadla přerušení, k vlastnímu přeplánování a nakonec k obnově kontextu vlákna, které se bude zpracovávat. Pokud je využito kooperativního typu plánování, dochází pouze k inkrementaci počítadla přerušení.

```
#pragma CODE_SEG __NEAR_SEG NON_BANKED
void interrupt vPortTickInterrupt( void )
{
    TC7 = TCNT + TempTC7;
    #if configUSE_PREEMPTION == 1
    {
        // Uložení kontextu, právě běžící úlohy
        portSAVE_CONTEXT();

        // Inkrementace počítadla ticku
        vTaskIncrementTick();

        // Přepnutí kontextu na jiné vlákno
        vTaskSwitchContext();

        // Nulování příznaku přerušení časovače
        TFLG1 = 128;

        // Obnovení kontextu nové úlohy
        portRESTORE_CONTEXT();
    }
    #else
    {
        vTaskIncrementTick();
        // Nulování příznaku přerušení časovače
        TFLG1 = 128;
    }
    #endif
}
#pragma CODE_SEG DEFAULT
```

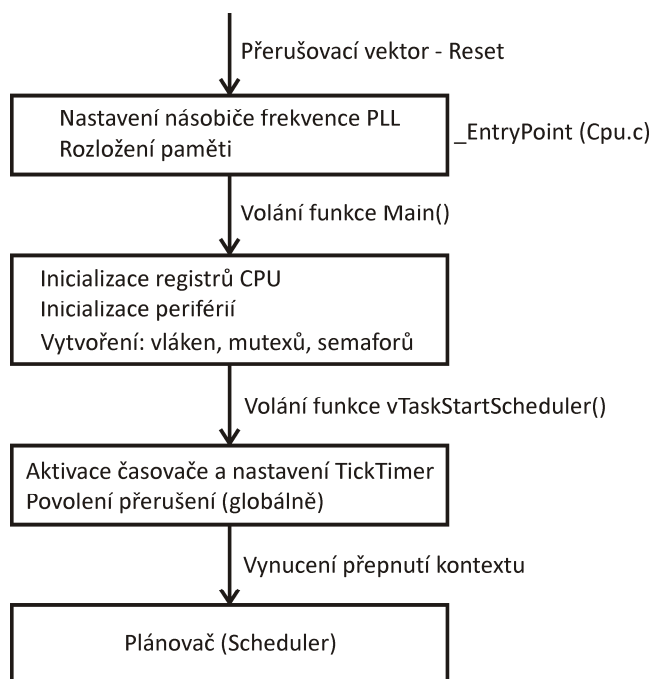
Tab. 10 Výpis programu – obsluha přerušení časovače (plánování vláken)

7.3.3 Algoritmus spouštění operačního systému

Pro úplnost je uveden postup, jak vlastně dojde ke spuštění plánovače, a tím i operačního systému. Na obrázku Obr. 30 vidíme algoritmus, který bude dále popsán.

Po přivedení napájecího napětí k mikroprocesoru se program začne vykonávat od adresy vektoru přerušení definovaného pro reset. V našem případě se nachází v souboru *Cpu.c*, ve kterém se provede nastavení frekvence sběrnice procesoru a inicializace paměti. Poté se již začne vykonávat uživatelská část programu, jako například vytvoření jednotlivých vláken, mutexů a front zpráv. V této chvíli námi definovaná vlákna ještě neběží.

Voláním funkce „*vTaskStartScheduler()*“ mimo jiné dojde k povolení přerušení od periférií a k aktivaci časovače spouštějící plánovač. Na konec se vyvolá softwarové přerušení, které zajistí první spuštění plánovače, ještě před jeho aktivací časovačem, dojde tedy k takzvanému vynucenému přepnutí kontextu. Při správném chodu by se program nikdy neměl vrátit zpět do funkce „*main()*“, ze které jsme plánovač spustili.



Obr. 30 Spouštění operačního systému

7.3.4 Měření provedená na operačním systému

Na operačním systému FreeRTOS je provedeno několik časových měření za účelem zjištění vytížení mikroprocesoru při zpracování systémových operací, jako jsou:

- Doba přepnutí kontextu
- Doba zamknutí a odemknutí mutexu
- Zápis do fronty a čtení z fronty zpráv
- Doba pro vytvoření vlákna

Mikroprocesor MC9S12DP512, s vnitřní frekvencí sběrnice 25MHz, byl testován s verzí FreeRTOS 6.0.2. V tabulce níže jsou uvedeny výsledné doby trvání a jejich procentuální vyjádření vztažené k délce trvání systémového ticku o délce 1 ms. Tím dostaneme velikost režie operačního systému vyjádřenou v procentech.

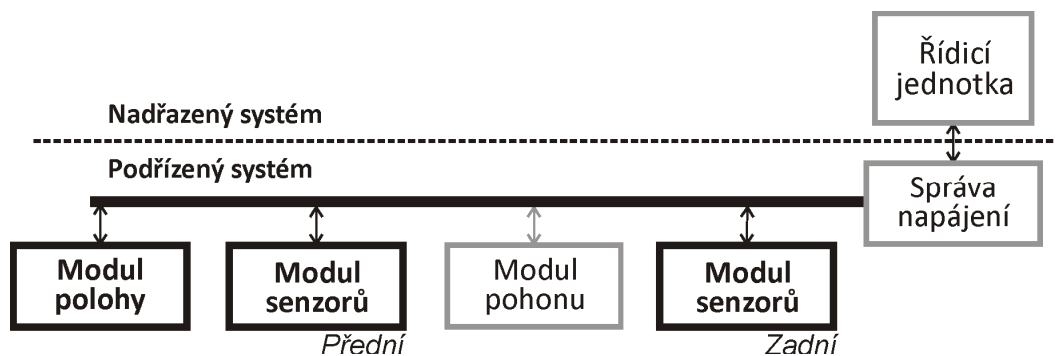
Operace	Doba provedení [μ s]	% času procesoru
Přepnutí kontextu	$8,25 \div 16$	$0,82 \div 1,6$
Vytvoření vlákna	158,8	15,9
Zamknutí mutexu	$22,55 \div 6,15$	$2,25 \div 0,62$
Odemknutí mutexu	$6,95 \div 15,8$	$0,7 \div 1,6$
Zápis do fronty zpráv	$22,2 \div 40,2$	$2,2 \div 4$
Čtení z fronty zpráv	$15,4 \div 19,4$	$1,5 \div 2$

Tab. 11 Časové měření na operačním systému FreeRTOS

Z naměřených výsledků je zřejmé, že nedochází k rapidnímu nárůstu režie při použití operačního systému FreeRTOS. Časové nároky systémových operací jsou minimální.

8 Komunikace mezi moduly na vozidle

Jelikož se řídicí systém vozidla rozkládá na několika modulech, jedná se tedy o řídicí systém distribuovaný, je spolehlivost a předvídatelnost komunikace velice důležitá. Propojení jednotlivých zařízení na vozidle je znázorněno na následujícím obrázku (Obr. 31).



Obr. 31 Schéma propojení jednotlivých zařízení vozidla

Vozidlo můžeme logicky rozdělit na dvě části, na systém nadřazený a podřízený. Na vozidle se nacházejí dvě různé komunikace. Podřízený systém je složen z několika modulů, které pro svou činnost musí navzájem komunikovat, a zároveň je nutné propojit obě části navzájem.

8.1 Výběr typu komunikace

Jak již bylo zmíněno, podřízený systém se skládá z několika modulů a bylo proto nutné vybrat jak budou navzájem propojeny. V následující tabulce jsou shrnuty různé typy komunikací a jejich výhody a nevýhody pro nasazení na vozidle.

Typ komunikace	Výhody	Nevýhody
IIC	Levná	Menší odolnost proti rušení
SPI	Vysoká přenosová rychlost	Menší odolnost proti rušení Větší počet vodičů
RS232	Jednoduchá	Menší odolnost proti rušení
LIN	Levná	Malá přenosová rychlost
CAN	Vysoká odolnost proti rušení	
USB	Vysoká odolnost proti rušení	Drahá
Ethernet	Vysoká odolnost proti rušení	Drahá

Tab. 12 Porovnání typů komunikací

Je vybrána sběrnice CAN, která se mimo jiné vyznačuje relativně vysokou přenosovou rychlostí, dobrou odolností proti rušení, nadprůměrnou spolehlivostí, nízkou cenou komunikačních obvodů, dobrou podporou ze stran různých výrobců, a také předvídatelností a prioritizací přenášených dat.

8.1.1 Sběrnice CAN

CAN je sériová komunikační sběrnice vyvinutá firmou Bosch, jejíž dobré vlastnosti zapříčinily její rozšíření i do mnoha dalších oborů, než pro které byla původně vyvinuta.

Mezi základní vlastnosti sběrnice patří:

- Sběrnice typu multimaster se sběrníkovou topologií
- Nody nemají specifické adresy. Místo adres se používají identifikátory zpráv
- Počet nodů může být dynamicky měněn (bez vlivu na ostatní nody)
- 1Mb/s při délce maximálně 40m
- Délka zprávy je maximálně 8 bajtů
- Metoda přístupu na síť – CSMA/CD w/ AMP

Přenos dat na sběrnici je zabezpečen několika způsoby:

- Cyclic Redundancy Check Error – Kontrolní součet (CRC)
- Form Error – Chyba formátu zprávy
- StuffError – Chyba vkládání bitů
- ACK Error – Chyba potvrzení
- Bit Error – Chyba přijatého bitu

Pro úplnost ještě uvedeme tvar datového rámce, ten je zobrazen na obrázku Obr. 32 . Mezi jeho hlavní části patří 11 nebo 29bitový identifikátor. Bit RTR (Remote transmission request), tedy žádost o přenos, sloužící k vyžádání odvysílání zprávy s příslušným identifikátorem. Bit IDE (Identifier extension) určuje, zda se jedná o zprávu se standardním 11bitovým, nebo rozšířeným 29bitovým identifikátorem. 4 bity DLC určující délku datového pole v bajtech, což je maximálně 8 bajtů. Za datovým polem je pak pole CRC.



Obr. 32 Formát datové zprávy sběrnice CAN

[14]

8.1.2 Implementace sběrnice CAN

Mikroprocesor Freescale MC9S12DP512 integruje rozhraní MSCAN, které podstatně zjednodušuje implementaci funkcí pro práci se sběrnici CAN. Pro inicializaci, vysílání a příjem jsou vytvořeny funkce využívající API funkcí operačního systému FreeRTOS.

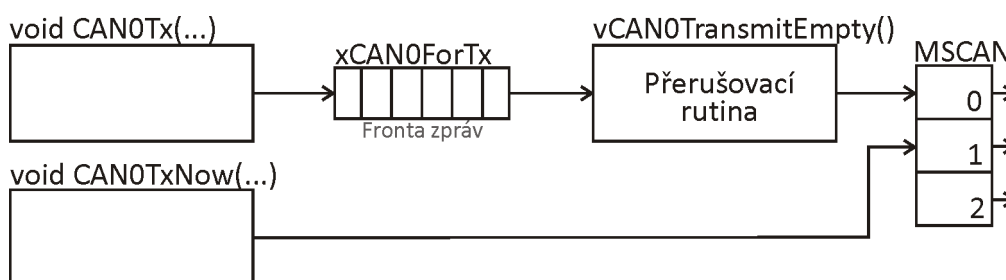
Inicializace sběrnice

Pro iniciaci sběrnice se využívá funkce `void CAN0Init(...)` jejímiž vstupními parametry jsou: přenosová rychlost, velikost fronty pro vysílání a ukazatel na strukturu nastavení vstupních filtrů řadiče sběrnice.

Funkce nastaví řadič MSCAN do inicializačního módu, ve kterém se mimo jiné provede nastavení požadované rychlosti a filtrace přichozích zpráv. Po přepnutí do standardního módu se povolí přerušení od události přijmutí nové zprávy a přetečení přijímací FIFO paměti. Přerušení od události prázdného vysílacího registru je prozatím zakázáno. Nakonec dojde k vytvoření front pro vysílání a pro příjem.

Vysílání dat na sběrnici

Na následujícím obrázku (Obr. 33) je znázorněno vysílání zpráv na sběrnici CAN. Pro standardní vysílání slouží funkce `void CAN0Tx(...)`, která má jako vstupní argument ukazatel na strukturu s daty pro odeslání. Jejím voláním se do fronty zpráv vytvoří kopie dat a povolí se přerušení od události prázdného vysílacího registru. Téměř ihned se vyvolá přerušovací rutina `vCAN0TransmitEmpty()`, která data z fronty uloží do vysílacích registrů řadiče MSCAN, a na konci rutiny se přerušení opět zakáže. Tato funkce má blokuující charakter, to znamená, že v případě zaplnění vysílací fronty se vláknu, které se v tomto momentě snaží vysílat, nebude přidělován procesorový čas do té doby, dokud se ve frontě místo neuvolní.

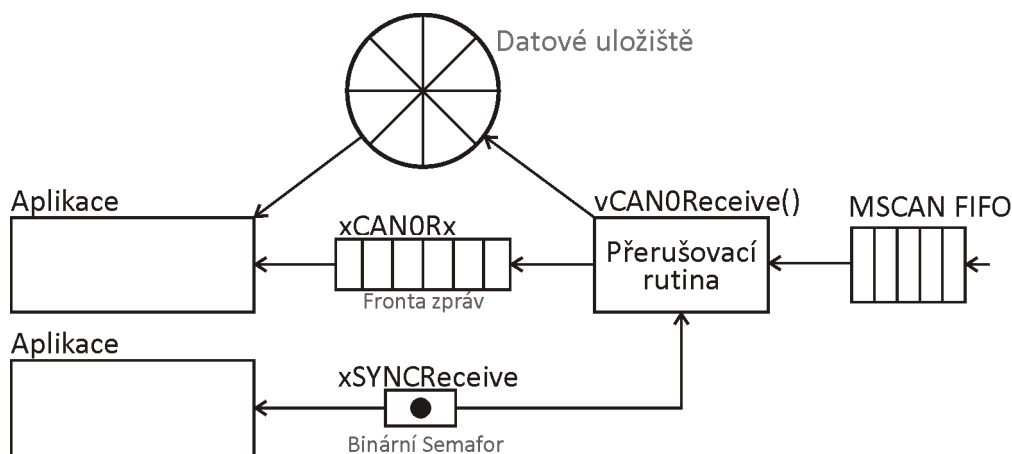


Obr. 33 Znázornění vysílání zpráv na sběrnici CAN

Druhý způsob odeslání dat je prostřednictvím funkce `void CAN0TxNow()`, která nevyužívá žádnou vyrovnávací paměť, ale rovnou přistupuje k vysílacím registrům řadiče MSCAN, a zároveň nastavuje pro vysílací registry vyšší prioritu, než kterou mají registry pro vysílání standardní cestou. Tím se zajišťuje okamžité vyslání požadovaných zpráv. Tato funkce má také blokuující charakter, ale na rozdíl od standardní cesty je vysílacímu vláknu i nadále přidělován procesorový čas, který je spotřebováván na kontrolu uvolnění vysílacích registrů.

Příjem dat ze sběrnice

Zprávy přijaté ze sběrnice procházejí v řadiči MSCAN vyrovnávací paměti typu FIFO a vyvolávají přerušení od události pro příjem. V přerušovací rutině se data uloží do datového úložiště a do fronty zpráv se odešle na tyto data pouze ukazatel. Je to z důvodu co nejmenší rezie v přerušení. Informace o volných místech v datovém úložišti je uchovávána, a pokud přerušovací rutina zjistí, že data není kam uložit, zakáže se přerušení od příjmu a aktuálně přijaté data ponechá v přijímacím registru. Tím se dosáhne toho, že se žádné přijaté zprávy „neztratí“, protože se začne naplňovat vyrovnávací paměť řadiče MSCAN po jejíž naplnění se příchozí zprávy nebudou potvrzovat. Aplikace (vlákno) konzumující příchozí zprávy musí kontrolovat zaplnění přijímacího úložiště a popřípadě opětovně povolovat přerušení od příjmu.



Obr. 34 Znáznornění příjmu zpráv ze sběrnice CAN

Další možností při příjmu je nastavení přerušovací rutiny tak, aby reagovala na zprávu s určitým identifikátorem, při jejímž příjmu se inkrementuje binární semafor, který aplikaci umožní přesnější zjištění doby, ve které tato zpráva dorazila.

8.2 Výběr vyšší vrstvy sběrnice CAN

Při výběru vyšší vrstvy pro sběrnici CAN byly uvažovány tyto standardy:

- CAL
- CANopen
- DeviceNet
- SAE J1939
- Vlastní řešení

Z důvodů dobré dostupnosti specifikací a jejímu hojnému využití i v automobilovém průmyslu je vybrán komunikační standard CANopen. Ovšem před jeho implementací byl vytvořen vlastní jednoduchý protokol, který umožňoval práci a aspoň částečné testování částí vozidla ostatních kolegů. Po krátkém čase ale vozidlo pracovalo jen s komunikačním protokolem CANopen.

8.2.1 CANopen

CANopen je komunikační protokol založený na standardu aplikační vrstvy CAL (CAN Application Layer), který je podporován mezinárodním sdružením výrobců průmyslové techniky - CiA (CAN in Automation) pro komunikace převážně na sběrnici CAN, ale i na jiných. Specifikace CANopen spadá do aplikační vrstvy ISO/OSI modelu.

Slovník objektů

Slovník objektů (Object Dictionary) je jednou z nejdůležitějších částí protokolu. Každý objekt ve slovníku je přístupný jak ze strany sběrnice, tak ze strany samotné aplikace, a tvoří tak jakéhosi prostředníka. Jednotlivé objekty jsou přístupné prostřednictvím své unikátní 16bitové adresy (index), kterou lze ještě dále dělit pomocí 8bitového sub-indexu.

Typy komunikací jak je definuje specifikace

- PDO – Komunikace typu PDO (Process Data Object) se využívá pro přenos procesních dat, které mají vysokou prioritu. Identifikátor, délka a obsah takovéto zprávy je definována svým komunikačním a mapovacím parametrem ve slovníku objektů. Prostřednictvím PDO je možné data vysílat i přijímat
- SDO – Komunikací typu SDO (Service Data Object) můžeme přistupovat do celého slovníku objektů, za cenu nižší priority přenášených zpráv. Tento druh komunikace je typu server-klient, a dokáže přenášet i velké objemy dat. Také se využívá pro počáteční nastavení CANopen zařízení, což může být například změna komunikačních parametrů zpráv typu PDO
- SYNC – Tento typ komunikace slouží k periodickému vysílání synchronizační zprávy jeho producentem. Jedná se o základní hodinový signál sběrnice.
- EMCY – Komunikace typu EMCY (Emergency) zajišťuje vysílání zpráv o chybách vyskytujících se v zařízení. Při chybě se vyšle zpráva na sběrnici, která nese informaci, o jakou chybu se jedná. Tato zpráva může, ale nemusí být přijata jedním nebo několika EMCY konzumenty
- NMT – Tato komunikace (Network Management) je typu master-slave, přičemž jedno zařízení na sběrnici je typu master a ostatní jsou typu slave. Každé zařízení typu slave může být inicializováno, spuštěno, monitorováno, resetováno nebo zastaveno
- NMT Error Control – Tato komunikace se využívá pro kontrolu stavu jednotlivých zařízení na sběrnici. Existují dvě verze, přičemž na síti by měla být aktivní pouze jedna z nich. První je „Node Guarding Protocol“, ve kterém zařízení typu master oslovuje jednotlivé zařízení, které odpovídají svými stavy. Vše se musí uskutečnit v předem definovaných intervalech. Druhým typem je „Heartbeat Protocol“, ve kterém zařízení typu slave odesílají zprávy se svým aktuálním stavem s předem danou periodou. Master pak vyhodnocuje časy period, ve kterém zprávy přicházejí

Rozdělení identifikátoru CAN zprávy

Jako standardní typ zprávy se využívá CAN zpráva dle specifikace 2.0A, teda ta s 11bitovým identifikátorem. Celý identifikátor se v CANopen také označuje jako COB-ID (Communication Object Identifier).

Identifikátor je rozdělen na dvě pole. První tvoří čtyři horní bity (10 až 7), které se nazývají „Function Code“ a označují službu, která tuto zprávu využívá (NMT, SYNC, PDO, atd.). Zbylé bity pak tvoří druhou část označovanou jako „Node-ID“, neboli adresa zařízení.

Následující tabulka (Tab. 13) shrnuje zprávy, které se na sběrnici mohou vyskytnout. První sloupec ukazuje pro jakou službu je daný rozsah používán, druhý pak ukazuje, jakou hodnotu mají jejich „Function Code“, a poslední pak zobrazuje rozsah identifikátoru.

Služba	Function Code (binárně)	Výsledný COB-ID
NMT	0000	0
SYNC	0001	128(80h)
TIMESTAMP	0010	256(100h)
EMERGENCY	0001	129(81h) – 255(FFh)
PDO1 (tx)	0011	385(181h) – 511(1FFh)
PDO1 (rx)	0100	513(201h) – 639(27Fh)
PDO2 (tx)	0101	541(281h) – 767(2FFh)
PDO2 (rx)	0110	769(301h) – 895(37Fh)
PDO3 (tx)	0111	897(381h) – 1023(3FFh)
PDO3 (rx)	1000	1025(401h) – 1151(47Fh)
PDO4 (tx)	1001	1153(481h) – 1279(4FFh)
PDO4 (rx)	1010	1281(501h) – 1407(57Fh)
SDO (tx)	1011	1409(581h) – 1535(5FFh)
SDO (rx)	1100	1537(601h) – 1663(67Fh)
NMT Error Control	1110	1793(701h) – 1919(77Fh)

Tab. 13 Zprávy protokolu CANopen (pole identifikátoru)

[6] [9]

8.2.2 Implementace CANopen

Pro komunikaci modulů na vozidle, nebylo třeba implementovat všechny části specifikace CANopen. Byly vytvořeny funkce a vlákna, které implementují tyto protokoly: PDO, SYNC a Heartbeat. Ve slovníku objektů jsou vytvořeny pouze nezbytné záznamy, které jsou popsány v následující tabulce.

Index	Název	Popis
1000h	DeviceType	Typ zařízení a jeho funkce
1001h	ErrorRegister	Místo pro ukládání vnitřních chyb zařízení
1005h	COBIDSyncMessage	Nastavení protokolu SYNC
1006h	CommCyclePeriod	Nastavuje periodu generování zprávy SYNC
1016h	ConsumHeartbeatTime	Nastavuje zařízení na Heartbeat konzumenta, přičemž definuje periodu vysílání pro jednotlivá zařízení na sběrnici
1017h	ProducerHeartbeatTime	Nastavuje periodu vysílání zprávy Heartbeat protokolu
1018h	IdentityObject	Obsahuje obecné informace o zařízení
1400 – 1407h	RPDOCommPar	Nastavuje jaké PDO zprávy se budou přijímat
1600 – 1607h	RPDOMappingPar	Nastavuje, jaké proměnné se v přijatých zprávách nacházejí
1800 – 1807h	TPDOCommPar	Nastavuje jaké PDO zprávy se budou vysílat
1A00 – 1A07h	TPDOMappingPar	Nastavuje, jaké proměnné jsou ve vysílaných zprávách

Tab. 14 Seznam nadefinovaných záznamů ve slovníku objektů

Inicializace

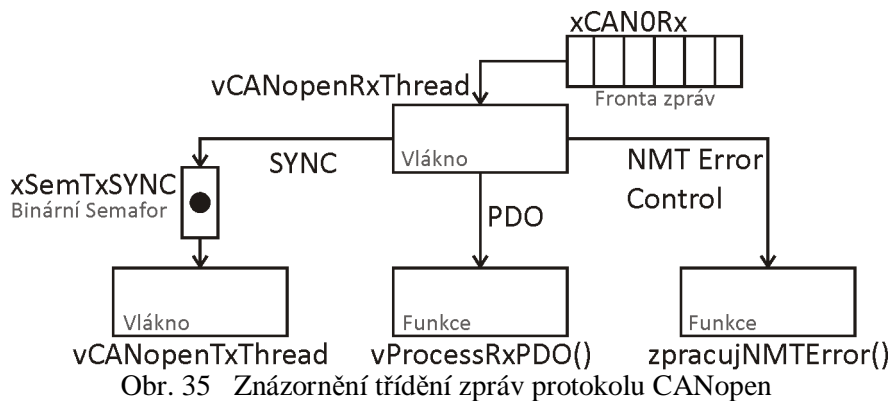
Inicializační funkce je nadefinovaná jako *void vCANopenInit(void)*, a měla by být provedena ještě před spuštěním plánovače operačního systému. Při inicializaci se provádí následující:

- Inicializuje se slovník objektů, tzn., nastaví se jeho defaultní parametry
- Vytvoří se synchronizační nástroje, potřebné pro jednotlivé protokoly
- Proveďte se inicializace řadiče sběrnice CAN
- Zjistí se, kolik RPDO komunikačních parametrů je nadefinováno. Ty se pak budou kontrolovat při příjmu PDO zprávy
- Zjistí se, kolik TPDO komunikačních parametrů je nadefinováno. Ty se pak budou vysílat
- Vytvoří se vlákno, které zajišťuje přebírání přijatých zpráv ze sběrnice CAN a jejich další třídění (*vCANopenRxThread*)
- Pokud je nadefinována alespoň jedna zpráva k vysílání, je vytvořeno vlákno obstarávající synchronní vysílání zpráv (*vCANopenTxThread*)
- Pokud je v objektu 1005h nadefinováno generování zprávy SYNC, vytvoří se pro její generování vlákno (*vCANopenTxSYNCThread*)
- Pokud má objekt 1017h nenulovou hodnotu (je nadefinována perioda vysílání) vytvoří se vlákno (*vCANopenHeartBeatProducer*) pro vysílání zpráv protokolu Heartbeat, v opačném případě se zařízení stává Heartbeat konzument a vytváří se příslušné vlákno (*vCANopenHeartBeatConsumer*)

Maximálně se tedy pro CANopen vytváří čtyři vlákna.

Příjem a třídění zpráv

Pro příjem a roztřídění příchozích zpráv je využito vlákno s názvem *vCANopenRxThread*. Třídění se provádí pomocí hodnoty identifikátoru zprávy, hlavně částí „Function Code“. Na následujícím obrázku (Obr. 35) je znázorněno, jak zprávy procházejí tříděním až ke svému zpracování.



Vlákno pro třídění čte data z fronty zpráv *xCAN0Rx*. Přijetí zprávy SYNC pro systém znamená pokyn k odeslání PDO zpráv s nastaveným typem vysílání na synchronní. To se zajišťuje inkrementací binárního semaforu *xSemTxSYNC*, který odesílání spustí. V případě přijetí zprávy typu PDO se vyvolá funkce *vProcessRxPDO()*, ve které se podle RPDO komunikačních a mapovacích parametrů (defaultně objekty 1400 až 1407h a 1600 až 1607h) uloží přijaté procesní data. Pokud je zařízení nastaveno jako konzument Heartbeat protokolu přijímá i zprávy typu NMT Error Control. Tyto zprávy zpracovává funkce *zpracujNMTError()*, ve které se ukládají časy přijetí zpráv od jednotlivých Heartbeat producentů.

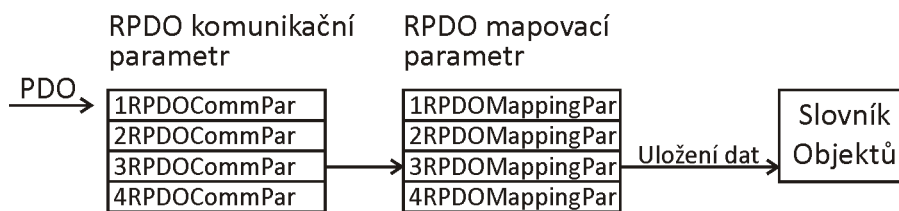
SYNC

K nastavení vysílání zprávy SYNC se využívají objekty 1005 a 1006h. V prvním objektu se vysílání aktivuje/deaktivuje a v druhém se nastavuje perioda vysílání v mikrosekundách. V tomto případě je nastavení periody omezeno na násobky milisekund s nejmenší nenulovou hodnotou jedné milisekundy (1000uS). Periodické vysílání zajišťuje vlákno *vCANopenTxSYNCThread*, které využívá funkce *CAN0TxNow(...)*. Ta nepoužívá vysílací frontu zpráv, ve které mohou být i zprávy s nízkou prioritou, ale přistupuje přímo k řadiči MSCAN, čímž je zachována co největší přesnost periody vysílání.

Při příjmu zprávy SYNC se pouze inkrementuje semafor, na který čeká vlákno pro periodické (synchronní) vysílání PDO zpráv.

PDO

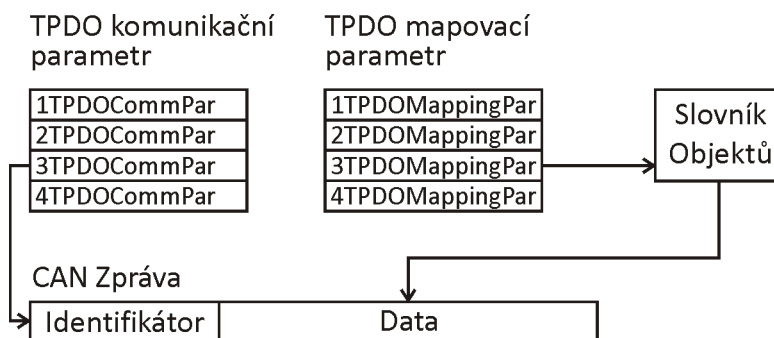
Po přijetí zprávy typu PDO je vyvolána funkce *void vProcessRxPDO()*, ve které se rozhodne, jestli je přijatá zpráva určena právě pro toto zařízení. To jak funkce rozhoduje, zda se má zpráva přijmout, je nastíněno na následujícím obrázku (Obr. 36). Shoda identifikátoru zprávy s RPDO komunikačním parametrem znamená, že zpráva je určena pro toto zařízení. Následně se vyhodnotí RPDO mapovací parametr, který nese informace o počtu a rozložení jednotlivých proměnných ve zprávě, které se následně uloží na své pozice ve slovníku objektů.



Obr. 36 Způsob dekódování přijaté PDO zprávy

CANopen specifikuje několik typů událostí pro vysílání PDO zpráv. Pro potřeby vozidla bylo nutné implementovat vysílání na událost příjmu zprávy SYNC (synchronní) a také na události, které vznikají uvnitř systému.

Zařízení může kdykoli PDO zprávu vyslat voláním funkce *void vSendTxPDO(...)*. Tato funkce má vstupní argument index (ze slovníku objektů) odpovídající TPDO komunikačnímu parametru zprávy, která se má vyslat na sběrnici. Zpráva se složí z TPDO komunikačního parametru a proměnných v pořadí, které je určené TPDO mapovacím parametrem. Sestavení zprávy je ukázáno na následujícím obrázku (Obr. 37).



Obr. 37 Způsob odesílání PDO Zpráv

Synchronní vysílání zpráv zajišťuje vlákno *vCANopenTxThread*, které při příjmu zprávy SYNC prochází všechny TPDO komunikační parametry, ve kterých je určeno, zda a jak často (kolik SYNC zpráv musí přijít) se příslušná zpráva vysílá.

Pro přenos většího objemu procesních dat definuje CANopen takzvané multiplexované PDO, zkráceně MPDO. Existují dvě verze, ale pro potřeby robotického zařízení postačuje implementování DAM MPDO (Destination Adress Mode PDO), které sebou nese jak data, tak i adresu (index, subindex) objektu v cílovém zařízení, kde se přenášené data mají zapsat. Přijímání DAM MPDO probíhá automaticky, pokud je DAM MPDO nadefinováno v příslušném RPDO komunikačním parametru. Pro vysílání se využívá funkce *void vSendDAM_MPDO(...)*.

Heartbeat Protokol

Pokud je zařízení nastaveno jako producent Heartbeat zpráv (objekty 1016 a 1017h), je spuštěno vlákno *vCANOpenHeartBeatProducer*, ve kterém se s nastavenou periodou vysílají na sběrnici zprávy informující mastera (zařízení pracující jako Heartbeat konzument) o aktuálním stavu zařízení.

V případě, že je zařízení nastaveno na konzumenta Heartbeat zpráv je spuštěno vlákno *vCANOpenHeartBeatConsumer*, ve kterém se periodicky kontrolují odezvy jednotlivých zařízení na sběrnici. Pokud nastane situace, kdy od jednoho zařízení nepřijdou dvě po sobě jdoucí zprávy v nastaveném časovém intervalu, vyhodnotí se to jako chyba vysílání tohoto zařízení, a voláním funkce *void ProcessHeartBeatError(...)* se dá aplikaci vědět, které zařízení má problémy.

8.3 Komunikace s hlavní řídicí jednotkou

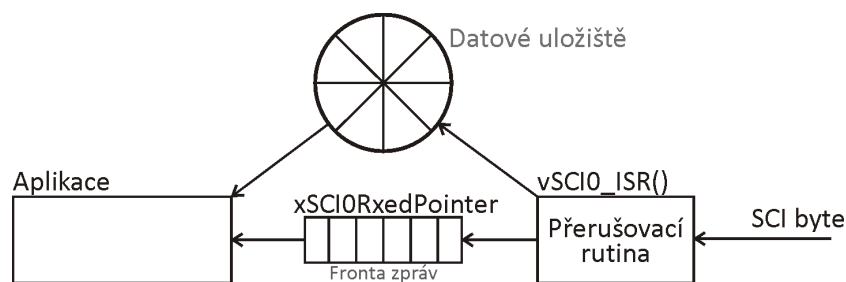
8.3.1 Výběr komunikace

Při výběru komunikačního rozhraní hlavní řídicí jednoty s modulem správy napájení existovaly tyto možnosti: sériový port, ethernet, SPI a USB. Jelikož hlavní jednotka používá jeden ze svých několika sériových portů pro spojení se vzdáleným vizualizačním systémem, pro který už byl vytvořen komunikační protokol, bylo opět výhodné zvolit sériové rozhraní a použít již vytvořený komunikační protokol.

8.3.2 Implementace rutin pro příjem a vysílání

Pro inicializaci rozhraní sériového portu byla vytvořena funkce *void vSCIO_init(...)*. Její vstupními parametry jsou hodnoty požadované rychlosti a hloubka fronty zpráv pro vysílání.

Příjem i vysílání dat pracuje podobně jako u rozhraní sběrnice CAN. U příjmu dat je rozdíl v tom, že ukazatel na přijatá data se odesílá až po příjmu a sestavení celé zprávy (v obslužné rutině přerušení).



Obr. 38 Znáznornění příjmu zpráv sériovým portem

Pro odesílání znaků, je využito fronty zpráv a přerušení. Aplikace data odesílá do fronty, a přerušovací rutina data z fronty vysílá. Funkce pro vysílání má název *void vSCIO_send(...)*.

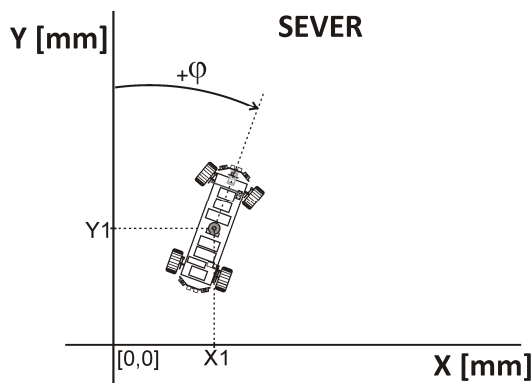


Obr. 39 Znáznornění vysílání dat ze sériového portu

9 Určování polohy v prostoru

9.1 Mapové podklady

Prostor, ve kterém se robotické zařízení pohybuje, je zakreslen v geometrických mapách v kartézském souřadném systému. Základní jednotky mapy jsou milimetry. Mapa je uložena v hlavní řídicí jednotce a modul polohy pouze udržuje informaci o poloze a natočení zařízení v této mapě.



Obr. 40 Souřadnicový systém mapových podkladů

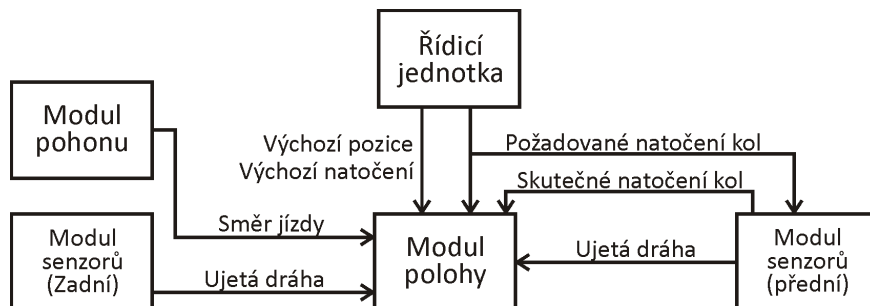
Kladná poloosa osy X směřuje na východ a kladná poloosa osy Y směřuje na sever. Úhel natočení se určuje po směru hodinových ručiček od kladné části osy Y, a nabývá hodnot z intervalu $< 0,360$).

Aby bylo možné určovat i polohu pomocí navigačního systému GPS, je počátek mapy označen její polohou právě v souřadnicích GPS.

9.2 Odometrie

Při výpočtu relativní změny polohy z odometrie jsou k dispozici údaje o ujeté dráze všech čtyřech kol, požadovaném a skutečném natočení předních kol, směru pohybu (dopředu/dozadu), natočení zařízení a aktuální poloze v souřadnicovém systému.

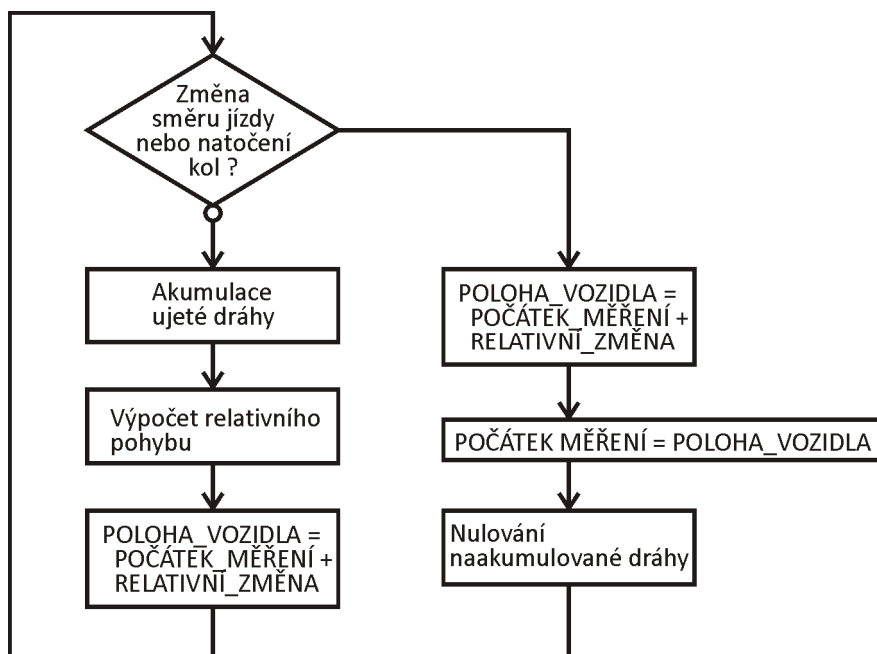
Na následujícím obrázku (Obr. 41) je znázorněn tok dat mezi moduly zařízení, které jsou využity pro výpočet. Data jsou opakovaně přenášena po sběrnici s taktem 100ms.



Obr. 41 Znázornění toků dat při výpočtu polohy z odometrie

Hodnotu výchozí pozice a výchozího natočení používá hlavní řídicí jednotka pro nastavení pozice po spuštění zařízení, nebo k aktualizaci naší pozice během provozu, pokud zjistí aktuální polohu jiným způsobem (například z map).

Na následujícím algoritmu (Obr. 42) je ukázáno, jak je relativní pohyb vypočítáván. Kdybychom každých 100ms vypočetli změnu polohy za tento čas a přičítali bychom ji okamžitě k aktuální poloze, nastávala by situace, kdy ujetá dráha by byla tak malá, že by mohlo docházet k numerickým chybám při výpočtech. Z tohoto důvodu je použit algoritmus, ve kterém se ujetá dráha akumuluje tak dlouho, doku nedojde ke změně směru jízdy, nebo natočení kol. Je to z toho důvodu, že při těchto změnách dochází i ke změně poloměru kružnice, které vozidlo bude opisovat. Během jízdy je vypočítáván relativní pohyb z celkové dráhy, které zařízení urazilo od poslední změny natočení kol, nebo směru jízdy. Až se jeden z těchto údajů opět změní, je relativní pohyb natrvalo přičten k naší pozici na počátku měření.

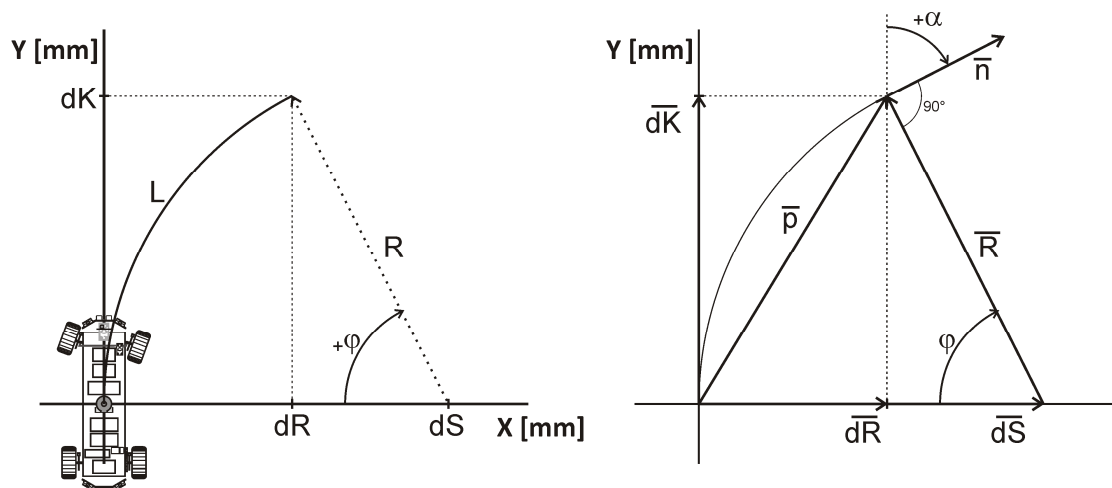


Obr. 42 Zjednodušený algoritmus výpočtu relativní změny polohy

Nejjednodušším pohybem pro výpočet relativní změny polohy je pohyb po přímce. Jelikož známe natočení vozidla a dráhu jím ujetou, známe tak i vektor na novou polohu v polárních souřadnicích. Polární souřadnice přepočteme do kartézských a složky x a y přímo odpovídají složkám relativní změny.

Dalšími případy jsou ty, kdy se vozidlo pohybuje po kružnici. Díky geometrii řízení na přední nápravě, vozidlo pokud jede vpravo, neopisuje kružnici o stejném poloměru, jako když jede vlevo (při shodném, ale opačném úhlu natočení řídicího servo-pohonu). Proto je nutné rozlišovat celkem čtyři směry pohybu (dopředu-doprava, dopředu-doleva, dozadu-doprava, dozadu-doleva). Poloměry jednotlivých kružnic byly získány měřením (pro šest různých úhlů natočení servo-pohonu, pro všechny čtyři směry).

Jednoduchý případ pohybu po kružnici nastává, pokud je vozidlo natočeno rovnoběžně s jednou z os souřadnicového systému. Takový případ je zobrazen na následujícím obrázku (Obr. 43). Vlevo je ukázána situace, jak probíhá pohyb při zatačení a co při něm měříme. Na pravé straně obrázku je nakreslená totožná situace, ale za pomoci vektorů.



Obr. 43 Výpočet pohybu po kružnici s nulovým počátečním natočením

Při pohybu měříme ujetou vzdálenost L , ze které za pomoci známého poloměru R opisované kružnice vypočteme úhel φ takto:

$$\varphi = \frac{L}{R} \quad [rad] \quad (9)$$

Složky pohybu dK a dR vypočteme z parametrického vyjádření kružnice takto:

$$\begin{aligned} dR &= R - R \cdot \cos(\varphi) \quad [mm] \\ dK &= R \cdot \sin(\varphi) \quad [mm] \end{aligned} \quad (10)$$

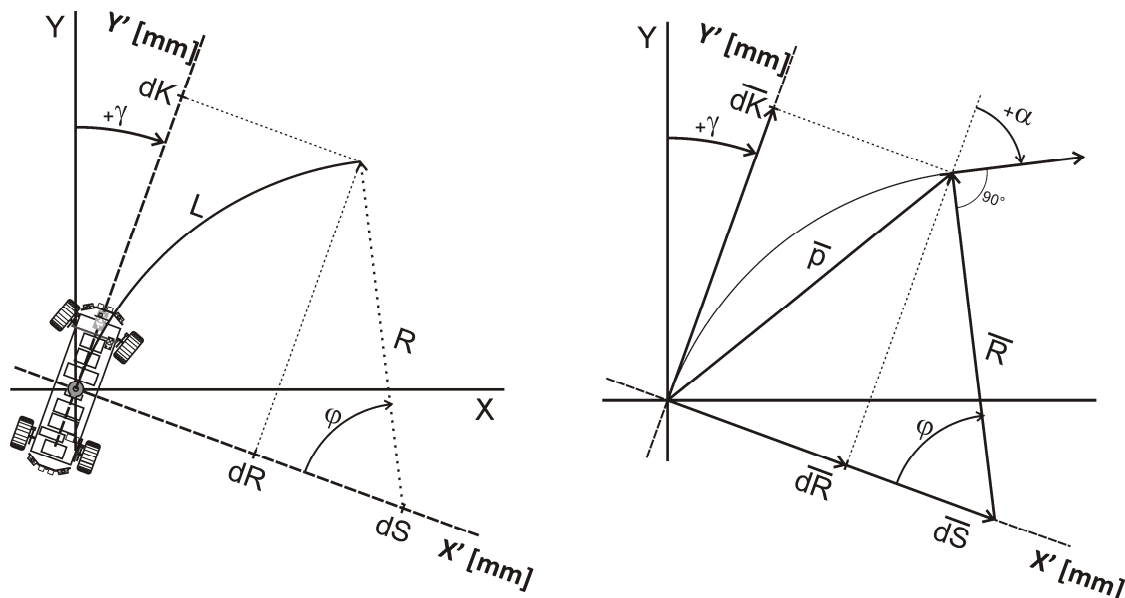
Vektory \overline{dR} a \overline{dK} vytvoříme díky znalosti jejich délek a jejich úhlů natočení. Pro vektor \overline{dR} je to 90° a pro \overline{dK} je to 0° , pokud je dK kladné a 180° , pokud je dK záporné. Sečtením obou vektorů vznikne vektor \overline{p} jehož koncový bod určuje naši novou polohu.

Díky znalosti poloměru kružnice R , je nám známa také souřadnice jejího středu dS . Vektor na střed kružnice \overline{dS} vypočteme z hodnoty jeho délky (dS) a úhlu natočení, který je totožný s úhlem vektoru \overline{dR} , tedy 90° . Rozdílem vektorů \overline{p} a \overline{dS} dostaneme vektor \overline{R} ze středu kružnice na naši novou polohu. Vytvořením kolmého vektoru na vektor \overline{R} získáme vektor \overline{n} , který má směr našeho nového natočení. Pro případ, kdy jsou přední kola stočena doprava, se kolmý vektor se správným směrem vytváří takto:

$$\begin{aligned} x_1 &= y_2 \\ y_1 &= -x_2 \end{aligned} \quad (11)$$

Kde vektor \overline{R} je dán uspořádanou dvojicí (x_2, y_2) a kolmý vektor \overline{n} je dán uspořádanou dvojicí (x_1, y_1) .

Pro obecný případ, kdy je zařízení na počátku natočeno libovolným směrem, se musí souřadnicový systém pro výpočet pootočit o úhel počátečního natočení γ tak, jak je znázorněno na následujícím obrázku (Obr. 44).



Obr. 44 Výpočet pohybu po kružnici s nenulovým počátečním natočením

Výpočet relativního pohybu i koncového natočení zařízení se provádí obdobně jako u předešlého modelového příkladu.

Ke změnám dochází při výpočtu vektorů \overline{dR} a \overline{dK} . Úhly, ze kterých se vektory vytváří, jsou dány vztahem:

$$\begin{aligned} \angle \overline{dR} &= \gamma + 90^\circ & [^\circ] \\ \angle \overline{dK} &= \gamma & [^\circ] \end{aligned} \quad (12)$$

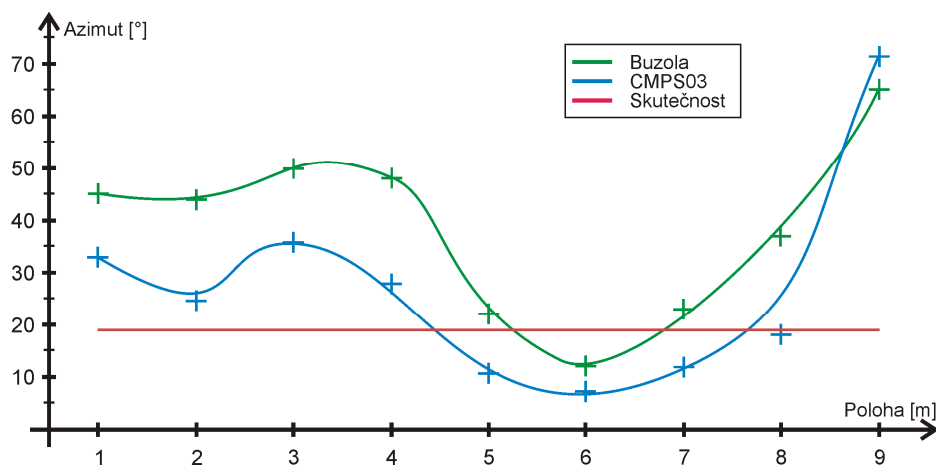
Pokud je hodnota dK z parametrického vyjádření záporná přičítá se k úhlu $\angle \overline{dK}$ ještě konstantní úhel 180° .

Pokud zařízení jede pouze rovně, bylo dosaženo relativní přesnosti určení polohy pod 0.5%. V případě, kdy zařízení jede pouze po jediném poloměru kružnice je relativní přesnost pod 1%. V ostatních případech, tedy kdy se střídají poloměry kružnic i pohyb rovně, je relativní přesnost určení polohy pod 10%.

9.3 Elektronický kompas

Měření magnetického pole země pro určení natočení zařízení v budově je problematické z důvodu výskytu lokálních zdrojů magnetického pole a železobetonové konstrukce budov.

Následující graf (Obr. 45) ukazuje závislost změřeného azimutu na poloze v místnosti. Bylo uskutečněno celkem devět měření, vzdálených od sebe 1 metr. Čára červenou barvou naznačuje, jaká je skutečná hodnota azimutu (bez uvažování magnetické deklinace). Zelený průběh je změřen poloprofesionální buzolou a modrý je změřen elektronickým kompasem CMPS03. Je patrné, že změřená hodnota silně závisí na poloze v místnosti. Maximální naměřená absolutní chyba byla 51° .



Obr. 45 Závislost změřeného azimutu na poloze v místnosti

Abychom omezili chybu vznikající magnetickým polem, které vytváří samo zařízení, je kompas umístěn nad zařízením na speciálním držáku a je uchycen pomocí materiálů z umělých hmot.

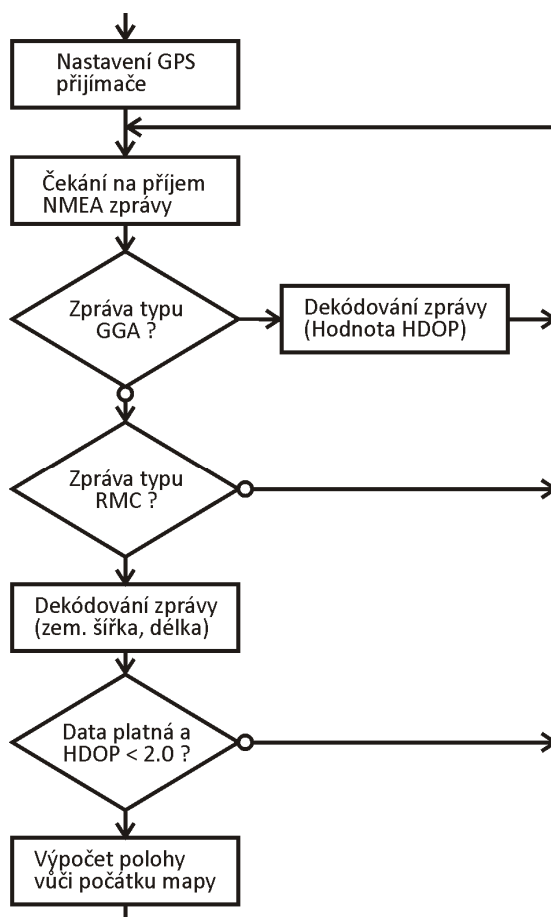
Výrobce elektronického kompasu udává maximální absolutní chybu měření (ve venkovním prostředí) 4° . Při měření kalibrační křivky, pro hodnoty 0 až 360° s krokem 5° ve směru hodinových ručiček i proti němu, byl naměřen rozdíl mezi jedním a druhým směrem měření při stejném natočení až $6,2^\circ$.

Ze změřených výsledků je jasné, že se tento kompas nedá použít uvnitř budov, a bude využíván pouze, pokud se zařízení bude pohybovat ve venkovním prostředí.

9.4 Navigační systém GPS

Navigační systém GPS je v podstatě určování polohy pomocí aktivních objektů v prostředí. Těmito objekty jsou umělé satelity země, které vysílají signály, pomocí kterých se vypočítává poloha na zemi. Tato poloha je určena úhlem zeměpisné šířky a délky. Zeměpisná šířka určuje polohu na povrchu Země směrem k severu nebo jihu od rovníku. Je to úhel, který svírá rovina rovníku s přímkou, procházející středem Země a příslušným bodem na povrchu Země. Označuje se řeckým písmenem φ . Zeměpisná délka pak určuje polohu na povrchu Země směrem na východ nebo západ od nultého poledníku. Je to úhel, který svírá rovina nultého poledníku s rovinou místního poledníku. Označuje se řeckým písmenem λ .

Poloha robotického zařízení se určuje ze dvou souřadnic. První souřadnicí je počátek mapy a druhou je aktuální pozice změřená GPS přijímačem. Jakým způsobem jsou data přijímána, dekódována a kdy se z nich počítá poloha vůči počátku mapy je znázorněno na následujícím obrázku (Obr. 46).

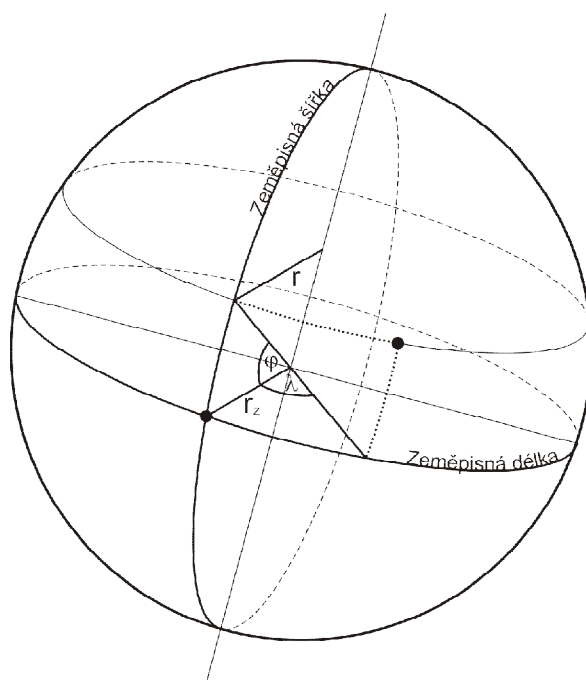


Obr. 46 Zjednodušený algoritmus výpočtu polohy z navigačního systému GPS

Po přivedení napájecího napětí k GPS přijímači je jeho komunikační rozhraní nastaveno na komunikační protokol NMEA, je tedy nutné pouze nastavit, aby přijímač vysílal zprávy typu RMC (Recommended minimum specific GPS/Transit data) a GGA (Global Positioning System Fix Data) s nejnižší možnou periodou, tedy s periodou jedné sekundy. Po té se už očekává příjem

jedné ze zpráv. Pokud se jedná o zprávu typu GGA, je dekodována a je z ní vyčtena hodnota HDOP (Horizontal Dilution of Precision), tedy hodnota, která vypovídá o přesnosti určení polohy v horizontálním směru. Pokud se jedná o zprávu typu RMC, je dekodována a pokud jsou přijaté data platná, jsou z ní vyčteny údaje o zeměpisné šířce a délce, azimutu a GPS času. Nakonec, pokud je aktuální hodnota HDOP menší než 2, je vypočítána poloha robotického zařízení vzhledem k počátku mapy.

Aby byla možná transformace GPS souřadnic do zvoleného souřadnicového systému, je nutné matematické popsání rozměrů planety Země. K tomu účelu se využívá popis pomocí takzvaného geoidu, referenčního elipsoidu anebo referenční koule. Pro malé vzdálenosti (desítky km) se dá s dostatečnou přesností využít aproximace pomocí referenční koule. Ta je znázorněna na následujícím obrázku (Obr. 47).



Obr. 47 Znázornění úhlů na referenční kouli

Úhel φ definuje zeměpisnou šířku v rozmezí -90 až 90° , a úhel λ definuje zeměpisnou délku v rozmezí -180 až 180° . Hodnota r_z je poloměrem referenční koule pro aproximaci Země. V tomto případě je to 6371.1km . Hodnota r je poloměrem kružnice (rovnoběžné s rovinou kružnice opisující rovník), pro určitý úhel zeměpisné šířky φ .

Při určování polohy vůči počátku mapy se nejprve vypočte rozdílový úhel mezi aktuální polohou a počátkem mapy ve směru zeměpisné šířky ($\Delta\varphi$) i délky ($\Delta\lambda$). Poté se vypočte poloměr kružnice r :

$$r = 6371100000 \cdot \cos(\varphi) \quad [\text{mm}] \quad (13)$$

Z hodnot rozdílových úhlů, poloměru referenční koule r_z a poloměru kružnice r se vypočítá poloha v souřadnicovém systému takto:

$$X = \Delta\lambda \cdot r \quad [mm] \quad (14)$$

$$Y = \Delta\varphi \cdot 6371100000 \quad [mm] \quad (15)$$

Následující tabulka (Tab. 15) porovnává měření vzdálenosti ve srovnání s jinými kalkulátory. Pro výpočet byly použity tyto souřadnice:

- 49° 42' 51.784"N, 17° 54' 40.326"E
- 48° 55' 33.211"N, 17° 54' 40.326"E

Vypočtená vzdálenost	Zdroj	Referenční těleso
87 677 741 mm	Robotické zařízení	Referenční koule
87 690 000 mm	http://boulter.com/gps/distance/	-
87 619 019 mm	http://www.csgnetwork.com/gpsdistcalc.html	-
87 692 929 mm	http://www.movable-type.co.uk/scripts/latlong-vincenty.html	Elipsoid WGS-84

Tab. 15 Srovnání výsledků měření s jinými kalkulátory vzdáleností

Bohužel se, až na jeden případ, nepodařilo zjistit, jaké aproximační tělesa jednotlivé kalkulátory využívají. Protože se maximální velikosti map předpokládají do jednotek kilometrů, je použití aproximace pomocí referenční koule dostatečné.

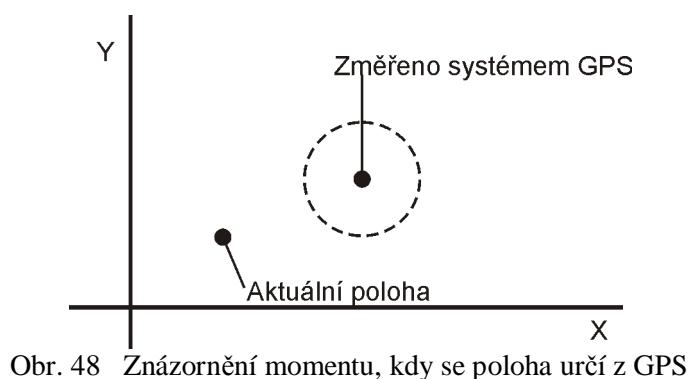
9.5 Kombinace dostupných údajů o poloze

K dispozici má robotické zařízení tyto údaje o poloze:

- **Azímút z elektronického kompasu** – tento údaj je využitelný pouze ve venkovních prostorách a přesnost klesá s výskytem rušivých magnetických polí
- **Azímút z odometrie** – dá se využít i uvnitř budov, ale jeho chyba roste s ujetou dráhou. Po čase (záleží na přesnosti měření) se tento údaj stává nepoužitelným. Je nutná jeho korekce
- **Azímút z GPS** – dá se využít jen ve venkovních prostorách, a jen při vyšších rychlostech
- **Poloha z GPS** – dá se využít jen ve venkovních prostorách, vykazuje velkou chybu určení polohy, ale velikost chyby neroste s časem
- **Poloha z odometrie** – dá se využít i uvnitř budov, ale jeho chyba roste s ujetou dráhou. Po čase (záleží na přesnosti měření) se tento údaj stává nepoužitelným. Je nutná jeho korekce

- **Upřesnění polohy hlavní řídicí jednotkou** – tato možnost je využita při spuštění zařízení a může se také využít k upřesnění polohy zařízení porovnáním mapy a okolního prostředí

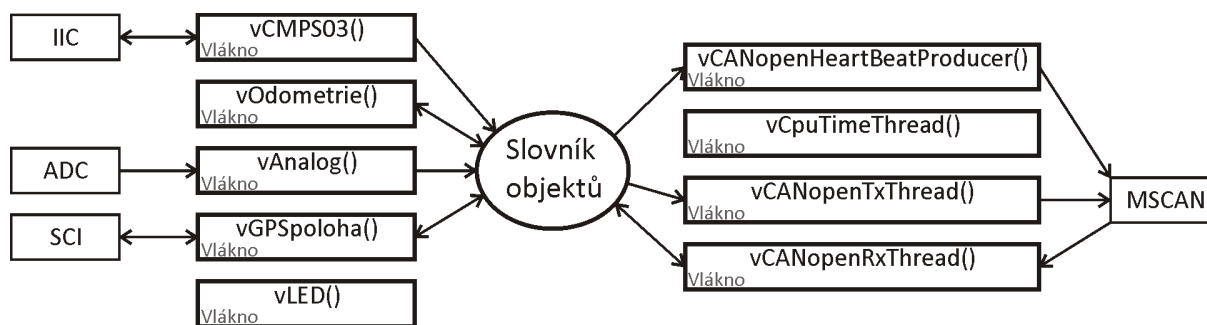
Je navrhnuo řešení, které pokud je dostupný signál ze systému GPS s dostatečnou velikostí hodnoty HDOP aktualizuje polohu zařízení, ale jen v případě pokud je odchylka takto změřené polohy od aktuální větší než dvacet metrů. Zároveň s aktualizací polohy se načte i nová hodnota natočení zařízení z elektronického kompasu. Uvnitř budovy se využívají pouze údaje z odometrie, případně upřesnění polohy hlavní řídicí jednotkou. Hodnota natočení zařízení změřená systémem GPS se nevyužívá, protože se zařízení nepohybuje dostatečnou rychlostí na to, aby tato hodnota byla dostatečně stabilní.



10 Popis software

10.1 Modul polohy

Software modulu polohy je vytvořen s využitím operačního systému FreeRTOS a je implementován do devíti vláken. Na následujícím obrázku (Obr. 49) jsou zobrazena jednotlivá vlákna, datové toky a periférie, které se využívají.



Obr. 49 Blokové schéma spuštěných vláken modulu polohy

Slovník objektů

Mimo standardních objektů potřebných pro komunikaci, jsou ve slovníku objektů nadefinovány i specifické proměnné. Ty uchovávají data, která jsou dostupná přes rozhraní CANopen. Jejich seznam se stručným popisem je v následující tabulce (Tab. 16).

Index	Název	Popis
0x2000	CANopenDummy	Pro mapování nechtěných částí PDO zpráv
0x2001	DrahaMSp	Ujetá dráha změřená modulem senzorů (přední)
0x2002	DrahaMSz	Ujetá dráha změřená modulem senzorů (zadní)
0x2003	NatoceniKol	Natočení předních kol zařízení
0x2004	SmerOtaceni	Směr pohybu zařízení
0x2005	Poloha	Poloha zařízení vůči počátku
0x2006	VychoziPoloha	Výchozí poloha zařízení v mapě
0x2007	G	Hodnoty z akcelerometru
0x2008	Plyn	Koncentrace hořlavých plynů
0x2009	Tlak	Změřená okolní tlak
0x200A	Natoceni	Natočení zařízení (Azimut)
0x200B	Cas	GPS čas v textovém formátu
0x200C	GpsPoloha	GPS poloha ve formátu 2*long
0x200D	ZacatekMapy	GPS poloha počátku mapy
0x200E	VychoziNatoceniVozidla	Výchozí natočení zařízení
0x200F	SkutecneNatoceniKol	Skutečné natočení předních kol

Tab. 16 Specifické objekty ve slovníku modulu polohy

Odometrie

Výpočet odometrie je zpracováván ve vláknu s názvem *vOdometrie()*. Vlákno čeká na příchod zprávy SYNC a na příchod dat, z těch pak vypočte relativní pohyb a aktualizuje polohu vozidla.

Elektronický kompas

Pro měření azimutu elektronickým kompasem je vytvořeno vlákno nazvané *vCMPS03()*. To je periodicky spouštěno každých 100ms. Pro komunikaci s kompasem se využívá sběrnice IIC.

GPS přijímač

Zpracování NMEA zpráv přijatých přes sériové komunikační rozhraní se provádí vlákem *vGPSpoloha()*. Zpráva po bajtech přijímána z GPS přijímače je v přerušení poskládána a uložena do vyrovnávací paměti. Tuto zprávu pak zmiňované vlákno dekoduje a do slovníku objektů uloží požadované informace. Pokud je potřeba provede se z GPS dat korekce polohy.

Indikace pomocí stavových LED

Pro indikaci stavu modulu je využito dvou luminiscenčních diod, červené a zelené barvy. Jejich činnost je řízena vlákem *vLED()*. Pokud je modul polohy v pořádku, střídavě se rozsvěcuje zelená a červená barva s periodou jedné sekundy.

Analogová měření

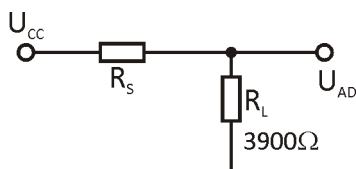
Pro analogová měření jsou využity dva analogově-digitální převodníky. Jedním převodníkem se měří hodnoty akcelerometru, ty jsou převedeny do znaménkového tvaru o velikosti jednoty bajtu a odeslány. Koncentrace výbušných látek a měření absolutního tlaku je měřeno druhým převodníkem. Pro měření je vytvořeno vlákno s názvem *vAnalog()*.

Výrobce tlakového čidla udává vzorec (3) pro výpočet výstupního napětí. Jeho úpravou za předpokladu, že napájecí napětí senzoru je stejné, jako kladné referenční napětí 10bitového AD převodníku dostaneme následující vztah:

$$P = \frac{\left(\frac{ADC}{1023} + 0.095\right)}{0.009} * 1000 \quad [Pa] \quad (16)$$

Kde P je absolutní tlak okolního vzduchu v jednotkách Pa a ADC je 10bitová hodnota změřená AD převodníkem.

Čidlo pro měření koncentrace výbušných látek reaguje na jejich různé druhy, a jelikož není známo, jaké látky se budou v okolí zařízení vyskytovat, je senzor nastaven pro měření koncentrace metanu, ostatní látky se měří s chybou. Na následujícím obrázku (Obr. 50) je nakresleno náhradní schéma zapojení tohoto senzoru. Odpor R_S je odpor senzoru a jeho velikost závisí právě na koncentraci snímaných látek. Na odporu R_L se měří napětí AD převodníkem, ze kterého se dopočítává hodnota odporu R_S .



Obr. 50 Náhradní schéma zapojení senzoru výbušných látek

Hodnota odporu R_S je získána ze vztahu:

$$R_S = \left(\frac{U_{CC}}{U_{RL}} - 1\right) \cdot R_L \quad [\Omega] \quad (17)$$

Ten se dá dále upravit:

$$R_S = \left(\frac{1023}{ADC} - 1\right) \cdot R_L \quad [\Omega] \quad (18)$$

Kde ADC je 11bitová hodnota změřená AD převodníkem.

Výrobce udává graf závislosti poměru R_S/R_0 ke koncentraci v jednotkách ppm, kde R_0 je odpor senzoru při koncentraci 1000ppm metanu. Tento graf byl přepracován na závislost poměru R_S/R_0 ke koncentraci v procentech, kdy 5% odpovídá 500ppm a 100% odpovídá 10000ppm.

Tato křivka byla aproximována funkcí:

$$M = -25 \cdot \log\left(\frac{R_s}{R_o} - 0.3\right) + 2 \quad [\%] \quad (19)$$

Kde M je koncentrace metanu v procentech (rozsah 0 až 100000ppm).

Využití výpočetního času mikroprocesoru

K sledování využití procesorového času, je používáno vlákno *vCpuTimeThread()* a *vApplicationIdleHook()*, které plánovač spouští jen tehdy, pokud ostatní vlákna nechtějí být zpracovávána. Vlákno *vApplicationIdleHook()* má nejnižší prioritu a je v něm neustále inkrementována hodnota proměnné *IdleHookCitac*. Z hodnoty této proměnné se periodicky každou sekundu ve vláknu *vCpuTimeThread()* nepřímou úměrou vypočítává vytížení procesoru v procentech. To znamená, že čím je větší načítaná hodnota, tím méně je procesor využíván.

U modulu polohy se hodnota využití procesoru pohybuje kolem 10%.

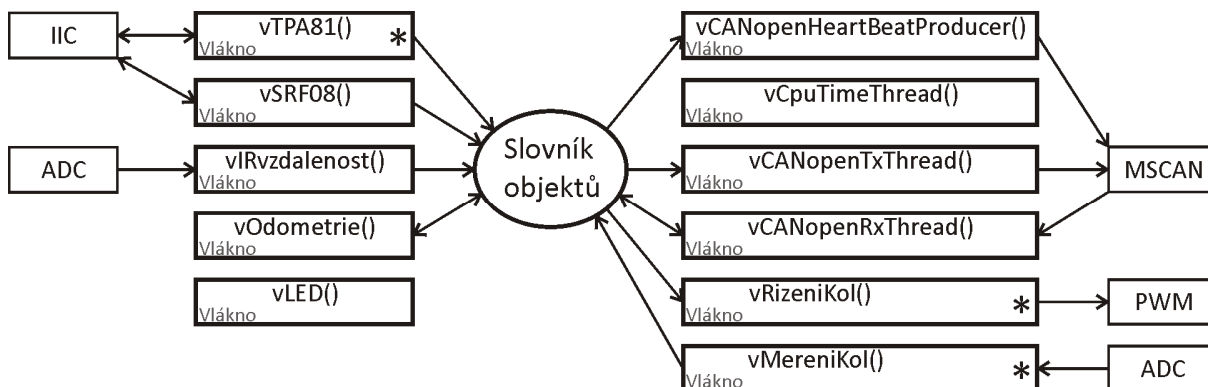
Popis souborů v projektu

Soubor	Popis
ATD.h; ATD.c	Analogově digitální převodník
can.h; can.c	Sběrnice CAN
CANopen.h; CANopen.c	CANopen – hlavní část komunikačního protokolu
Heartbeat.h; Heartbeat.c	CANopen – protokol Heart Beat
OD.h; OD.h	CANopen – slovník objektů
PDO.h; PDO.c	CANopen – protokol PDO
SYNC.h; SYNC.c	CANopen – protokol SYNC
Cpu.h; Cpu.c	Inicializace mikroprocesoru
Vektors.c	Nadefinování názvů přerušovacích rutin
GPS.h; GPS.h	Obsluha GPS přijímače ORCAM35
i2c.h; i2c.c	Sběrnice IIC
sci.h; sci.c	Sběrnice SCI
CpuTime.h; CpuTime.c	Měření využití výpočetního času mikroprocesoru
Generaldefs.h	Definování datových typů
Hardware.h; Hardware.c	Definice a inicializace rozhraní mikroprocesoru
Odometrie.h; Odometrie.c	Výpočet odometrie
main.c	Hlavní část programu

Tab. 17 Popis jednotlivých souborů použitých v projektu modulu polohy

10.2 Modul senzorů

Software modulu senzorů je vytvořen s využitím operačního systému FreeRTOS a je implementován do jedenácti vláken. Na následujícím obrázku (Obr. 51) jsou zobrazena jednotlivá vlákna, datové toky a periférie, které se využívají. Vlákna označená hvězdičkou nejsou na zadním modulu senzorů implementována.



Obr. 51 Blokové schéma spuštěných vláken modulu senzorů (přední)

Slovník objektů

Mimo standardních objektů potřebných pro komunikaci, jsou ve slovníku objektů nadefinovány i specifické proměnné. Ty uchovávají data, která jsou dostupná přes rozhraní CANopen. Jejich seznam se stručným popisem je v následujících tabulkách (Tab. 18 a Tab. 19). Slovníky předního a zadního modulu jsou rozdílné, protože v zadní části není třeba natáčet kola, a není zde osazeno termočláňkové pole.

Index	Název	Popis
0x2000	CANopenDummy	Pro mapování nechtěných částí PDO zpráv
0x2001	DrahaMSp	Ujetá dráha změřená modulem senzorů
0x2002	Ultrazvuk	Vzdálenost změřená ultrazvukovým čidlem
0x2003	NatoceniKol	Natočení předních kol zařízení
0x2004	IR	Vzdálenosti změřené infračervenými senzory
0x2005	Osvetleni	Osvětlení změřené čidlem na ultrazvuku
0x2006	Nadproud	Informace o nadproudu putujícího do modulu
0x2007	Teplota	Teplota změřená čidlem na termočláňkovém poli
0x2008	Tranzistory	Aktivace/deaktivace výstupních tranzistorů
0x2009	IRKamera	Změřený teplotní obraz
0x200A	SkutecneNatoceniKol	Skutečné natočení předních kol

Tab. 18 Specifické objekty ve slovníku modulu senzorů (přední)

Index	Název	Popis
0x2000	CANopenDummy	Pro mapování nechtěných částí PDO zpráv
0x2001	DrahaMSp	Ujetá dráha změřená modulem senzorů
0x2002	Ultrazvuk	Vzdálenost změřená ultrazvukovým čidlem
0x2004	IR	Vzdálenosti změřené infračervenými senzory
0x2005	Osvětlení	Osvětlení změřené čidlem na ultrazvuku
0x2006	Nadproud	Informace o nadproudu putujícího do modulu
0x2008	Tranzistory	Aktivace/deaktivace výstupních tranzistorů

Tab. 19 Specifické objekty ve slovníku modulu senzorů (zadní)

Odometrie

Měření ujeté vzdálenosti je synchronizováno s příchodem zprávy SYNC. Již v přerušení od příjmu se zálohuje počet impulzů načítaných do akumulátorů pro levé a pravé kolo. Hned poté je odblokováno vlákno *vOdometrie()*, ve kterém se z naměřených hodnot vypočte průměr a ten se odešle. Záloha načítaných hodnot se provádí v přerušení, aby v předním a zadním modulu byly shodné časové úseky měření.

Termočláňkové pole

Pohyb a vyčítání změřených hodnot z termočláňkového pole zajišťuje vlákno nazvané *vTPA81()*. Senzor je ovládán přes sběrnici IIC. Měření probíhá dokola v těchto krocích:

- Nastavení pozice
- Vyčkání 100ms
- Vyčtení změřených hodnot
- Odeslání naměřených hodnot dvěma MPDO zprávami

Celkem má senzor 31 poloh, a vždy na nulté pozici se navíc vyčítá hodnota okolní teploty.

Ultrazvukový dálkoměr

Měření vzdálenosti ultrazvukovým senzorem k případné překážce zajišťuje vlákno nazvané *vSRF08()*. Senzor je ovládán přes sběrnici IIC. Hodnota změřené vzdálenosti je ze senzoru vyčítána každých 65ms. Nejprve se pošle příkaz pro zahájení měření a po zmíněných 65 milisekundách se vyčte první „echo registr“ s výsledkem.

Natočení kol

K natáčení předních kol se využívá servo-pohon, který je řízen signálem PWM. Hlavní řídicí jednotka zasílá hodnotu v jednotkách stupňů, podle které se servo-pohon natáčí. Vše je řízeno vláknem *vRizeniKol()*.

Skutečné natočení servo-pohonu, tedy i kol, se měří z jeho zabudovaného potenciometru. Napětí změřené AD převodníkem je filtrováno plovoucím filtrem o délce 25 vzorků. Měření a filtrace se provádí ve vlákně *vMereniKol()*.

Infračervené dálkoměry

Měření vzdálenosti k případné překážce infračervenými senzory zajišťuje vlákno nazvané *vIRVzdalenost()*. Senzor má analogový výstup, který je periodicky měřen AD převodníkem. Výrobce udává převodní charakteristiku vzdálenosti k překážce vzhledem k výstupnímu napětí. Tento graf je aproximován funkcí:

$$L = \left(\frac{6000}{ADC} - 13 \right) \cdot 10 + 100 \quad (20)$$

Kde L je změřená vzdálenost v milimetrech a ADC je 11bitové číslo změřené AD převodníkem.

Využití výpočetního času mikroprocesoru

Sledování využití procesorového času, je provedeno shodně s modulem polohy. U předního modulu senzorů se hodnota využití procesoru pohybuje kolem 7%. U zadního je to kolem 6%.

Popis souborů v projektu

Soubor	Popis
ATD.h; ATD.c	Analogově digitální převodník
can.h; can.c	Sběrnice CAN
CANopen.h; CANopen.c	CANopen – hlavní část komunikačního protokolu
Heartbeat.h; Heartbeat.c	CANopen – protokol Heart Beat
OD.h; OD.h	CANopen – slovník objektů
PDO.h; PDO.c	CANopen – protokol PDO
SYNC.h; SYNC.c	CANopen – protokol SYNC
Cpu.h; Cpu.c	Inicializace mikroprocesoru
Vektors.c	Nadefinování názvů přerušovacích rutin
i2c.h; i2c.c	Sběrnice IIC
CpuTime.h; CpuTime.c	Měření využití výpočetního času mikroprocesoru
Generaldefs.h	Definování datových typů
Hardware.h; Hardware.c	Definice a inicializace rozhraní mikroprocesoru
RizeniKol.h; RizeniKol.c	Řízení servopohonu
main.c	Hlavní část programu
PA.h; PA.c	Akumulátory pulzů
TPA81.h; TPA81.c	Termočlánkové pole TPA81

Tab. 20 Popis jednotlivých souborů použitých v projektu modulu polohy

11 Závěr

Cílem této práce bylo navrhnout a vytvořit lokalizační a komunikační systémy robotického zařízení a ve spolupráci s ostatními kolegy pracujícími na projektu vytvořit jeho funkční prototyp. Zařízení je postaveno na zakoupeném podvozku modelu auta v měřítku 1:5, na kterém byly umístěny všechny jednotky potřebné k jeho funkci.

V rámci této práce byly celkem zhotoveny tři moduly. Dva z nich slouží jako sběrné body v přední a zadní části modelu. Mimo jiné k nim jsou připojeny senzory vzdálenosti a inkrementální snímače pro měření ujeté dráhy. Třetí modul vypočítává a udržuje informaci o aktuální poloze.

Na všech modulech robotického zařízení vyjma hlavní řídicí jednotky je implementován operační systém FreeRTOS, který usnadňuje vývoj aplikací a plně dostačuje pro naše potřeby. Tento operační systém je výrobcem stále vyvíjen a díky dobré dostupnosti zdrojových kódů je relativně jednoduché si přizpůsobit jeho funkce vlastním potřebám.

Poloha zařízení se v současné době určuje z navigačního systému GPS, měřením odometrie a azimutu určeného elektronickým kompasem. Zařízení také obsahuje mapu svého okolí s vyznačením možných tras pro dosažení cíle.

Systém odometrie dosahuje přesnosti určení polohy od 0.5 až po 10% a je závislá na trajektorii pohybu. Částečně je to způsobeno geometrií kol modelu vozidla, vůlemi v řízení, ale také poddimenzováním servo-pohonu pro řízení kol. Ten totiž v některých případech nedokáže otočit koly na požadovaný úhel, pokud se zařízení nehýbe.

Použitý přijímač GPS signálu se vyznačuje dobrou citlivostí a jeho implementace byla bezproblémová. Jsou vytvořeny funkce, pomocí kterých je jej možné nastavovat, a také dekodovat přijatá data. Jsou získávány informace o poloze, azimutu a přesnosti určení polohy v horizontálním směru. Poloha takto zjištěná je přepočítávána do jednotného souřadnicového systému pomocí aproximace Země referenční koulí.

Měřením bylo zjištěno, že se elektronický kompas v budovách nedá použít. Pracuje se s ním, pouze pokud se zařízení pohybuje ve venkovním prostředí.

Pro směřování absolutních a relativních údajů o poloze je vytvořen algoritmus, který porovnává aktuální pozici s pozicí změřenou systémem GPS.

Robotické zařízení je koncipováno jako distribuovaný systém řízení a pro komunikaci mezi moduly v zařízení je vybrána sběrnice CAN s aplikační vrstvou CANopen. Obě části komunikace (CAN, CANopen) jsou od sebe jednoznačně odděleny definovaným rozhraním. Při implementaci bylo využito API funkcí poskytovaných operačním systémem. Aplikační vrstva komunikace se snaží držet svých specifikací, i když některé protokoly nejsou obsaženy. Výhodnou takového přístupu je, že přizpůsobení jednotlivým modulům spočívá pouze v nadefinování parametrů a proměnných ve slovníku objektů, o vše ostatní se stará inicializační rutina při spuštění. Pro komunikaci s hlavní řídicí jednotkou je vybráno rozhraní sériového portu, pro které jsou implementovány rutiny pro příjem a odesílání dat. Na tomto základu je pak postavena aplikační vrstva. Pojednání o ní už není součástí této práce.

Do budoucna by pro směřování údajů o poloze bylo vhodné vytvořit systém využívající Kalmanova filtru, který bere do úvahy i neurčitosti měření jednotlivými metodami. Dále pro zpřesnění odometrie je nutné mechanicky zpevnit geometrii předních kol a nahradit stávající pohon silnějším s rychlejší odezvou.

Lokalizační systém vozidla je připraven na možné rozšíření sofistikovanějšími metodami určování polohy, protože hlavní řídicí jednotka má možnost kdykoli informace o poloze upravit. Absolutní údaj o poloze je k určování polohy bezpodmínečně nutný a jeho získání v prostorách budov je problematické. Bylo by vhodné doplnit na zařízení druhý laserový dálkoměr LMS100 a získat tak celkový obraz okolního prostoru a ten porovnávat s mapovými podklady. Neméně zajímavou možností k získání údajů o poloze je připojení širokoúhlé kamery a snímání stropu místností. Zpracováním takového obrazu stropu by bylo možné získat tvar, rozměry, ale také polohu v místnosti.

Závěrem lze říci, že i když se obor lokalizace v prostoru vyvíjí od pradávna, je v něm stále co vylepšovat a díky novým technologiím jsou zde možnosti přicházet s novými řešeními.

Literatura

- [1] NMEA Reference Manual [online]. San Jose, CA 95112 U.S.A. : [s.n.], Prosinec 2007 [cit. 2010-04-03]. Dostupné z WWW: <<http://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>>.
- [2] *Robot Electronics* [online]. [cit. 2010-04-03]. Dostupné z WWW: <<http://www.robot-electronics.co.uk/htm/tpa81tech.htm>>.
- [3] *Robot Electronics* [online]. [cit. 2010-04-03]. Dostupné z WWW: <<http://www.robot-electronics.co.uk/htm/cmeps3tech.htm>>.
- [4] BARRY, Richard. *Using the FreeRTOS real time Kernel : A Practical Guide*. [s.l.] : [s.n.], 2009. 164 s.
- [5] BARRY, Richard. *FreeRTOS Reference manual : API Functions and Configuration options*. [s.l.] : [s.n.], 2009. 119 s.
- [6] CAN in Automation. *CANopen Application Layer and Communication Profile : CiA Draft Standard 301*. [s.l.] : [s.n.], 2002. 135 s.
- [7] BORENSTEIN, L., EVERETT, H. R., FENG, L. *Where am I? : Sensors and Methods for Mobile Robot Positioning*. J. Borenstein. [s.l.] : [s.n.], 1996. 282 s.
- [8] Motorola, Inc. MC9S12DP256B : Device User Guide. 2nd edition. [s.l.] : [s.n.], 2005. 126 s.
- [9] PFEIFFER, Olaf, AYRE, Andrew, KEYDEL, Christian. *Embedded Networking with CAN and CANopen*. 1st edition. United States of America : RTC Books, San Clemente, CA, 2003. 535 s. ISBN 978-0-9765116-2-5.
- [10] KREJCAR, Ondřej. *Využití lokalizace uživatele pro prediktivní nahrávání dat v řídicích systémech*. Ostrava, 2008. 166 s. Dizertační práce. Vysoká škola báňská - Technická univerzita.
- [11] NEGENBORN, Rudy. *Robot Localization and Kalman Filters : On finding your position in a noisy world*. [s.l.], 2003. 143 s. Diplomová práce. UTRECHT UNIVERSITY.
- [12] *NMEA Messages in Orcam OEM-GPS Receivers* [online]. [s.l.] : [s.n.], 2009 [cit. 2010-04-03]. Dostupné z WWW: <<http://www.orcam.eu/res/default/orcamnmeamessage-summaryv2.pdf>>.
- [13] WINKLER, Zbyněk. *Robotika.cz* [online]. 2005 [cit. 2010-04-03]. Odometrie. Dostupné z WWW: <<http://robotika.cz/guide/odometry/en>>.
- [14] Controller area network In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2003, 2010 [cit. 2010-04-08]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Controller_area_network>.

- [15]Magnetic declination In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2003, 2010 [cit. 2010-04-08]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Magnetic_declination>.
- [16]*Orcam GPS35* [online]. [s.l.] : [s.n.], 2009 [cit. 2010-04-03]. Dostupné z WWW: <http://www.orcam.eu/res/default/gps35_pb_v20.pdf>.

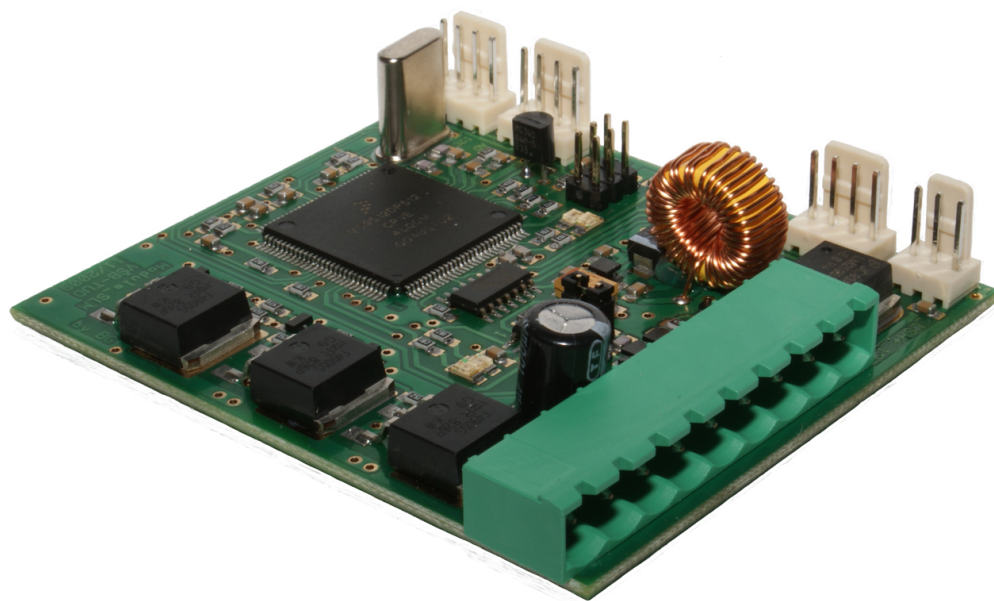
Seznam příloh

- I. Schéma zapojení modulu senzorů
- II. Fotografie modulu senzorů
- III. Schéma zapojení modulu polohy
- IV. Fotografie modulu polohy
- V. Dokumentace DPS inkrementálního senzoru CNY70
- VI. Popis PDO zpráv komunikačního protokolu CANopen
- VII. Fotodokumentace
- VIII. DVD-ROM

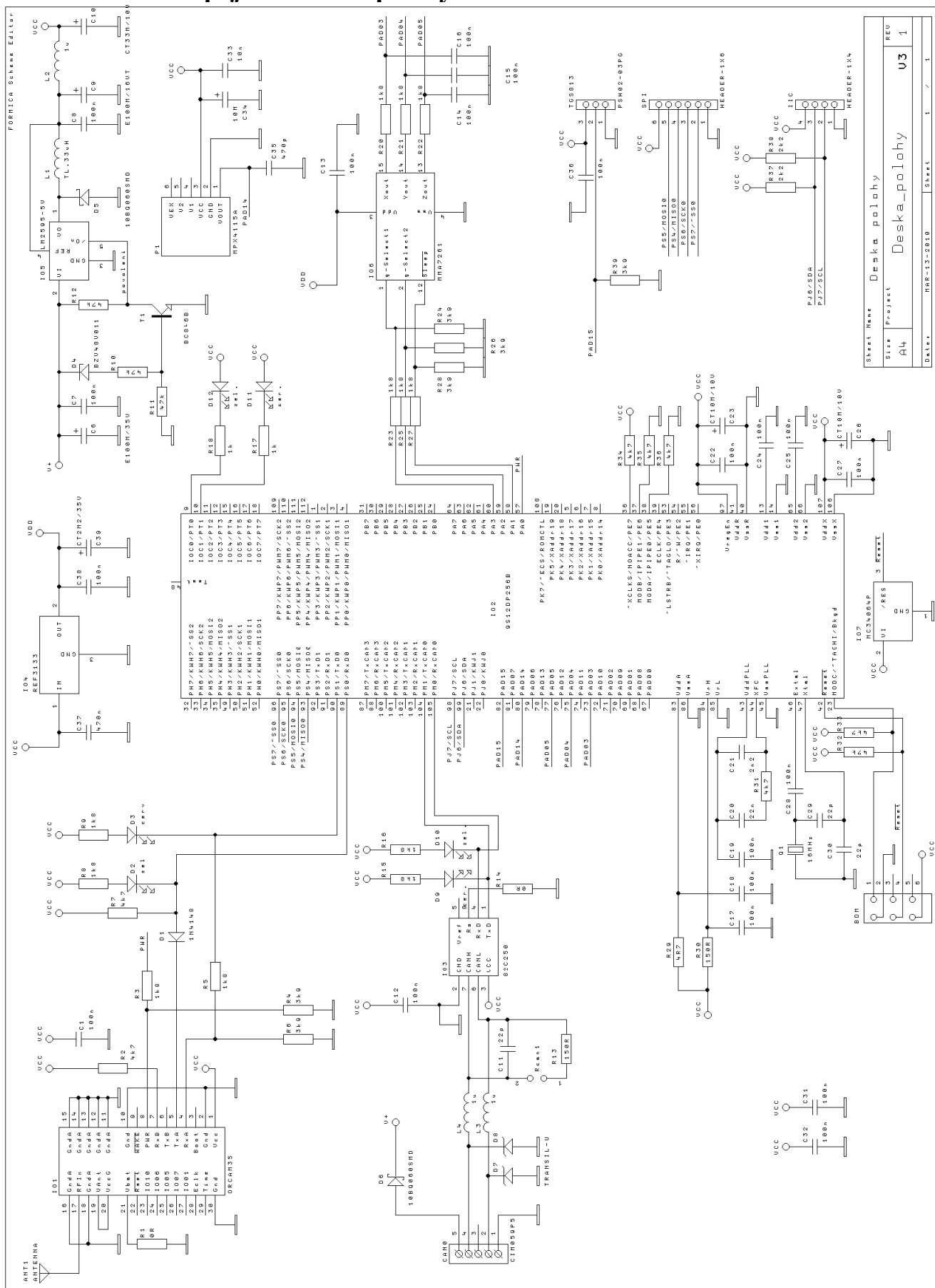
FORMICA Scheme Editor



II. Fotografie modulu senzorů

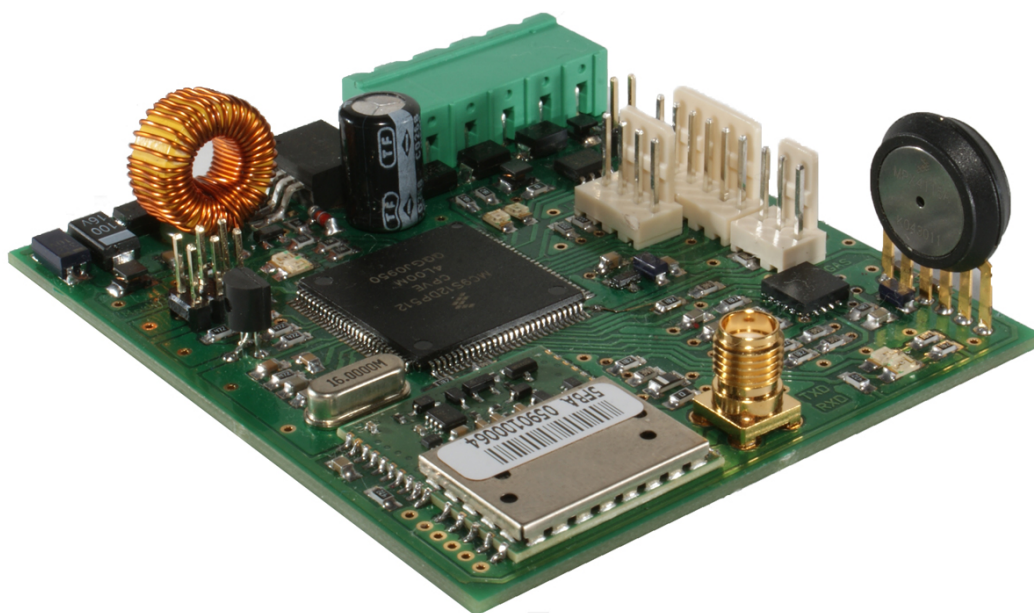


III. Schéma zapojení modulu polohy



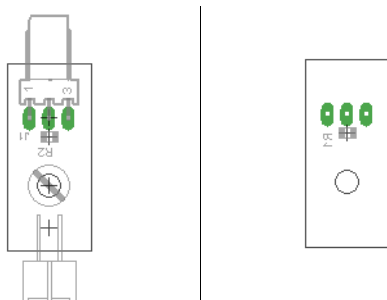
Sheet Name	Deska_polohy		REV
Size	A4	Deska_polohy	V3 1
Date:	MAR-13-2010	Sheet	1 / 1

IV. Fotografie modulu polohy

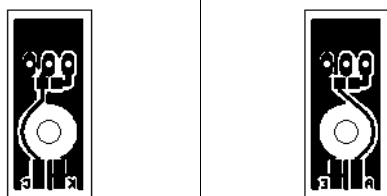


V. Dokumentace DPS inkrementálního senzoru CNY70

Osazení součástek (vlevo TOP, vpravo BOTTOM):



Vrstva mědi (vlevo TOP, vpravo BOTTOM):



VI. Popis PDO zpráv komunikačního protokolu CANopen

Zařízení	Zpráva	Identifikátor (COB-ID)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
Modul správy napájení	TPDO1 + BM	0x184	NabijeciNapeti [mV] 2001H/1		NabijeciProud [mA] 2001H/2		NapetiBaterie [mV] 2001H/3		OdebiranyProud [mA] 2001H/4		
	RPDO1 + MSp	0x202	Tranzistory [3bit] 2002H/1	Natočení kol [°] 2002H/2	VychodiNatoceniVozidla 2002H/3						
	RPDO1 + MSz	0x203	Tranzistory [3bit] 2003H/1	Natočení kol [°] 2003H/2							
	RPDO3 + MP	0x401	PocatekMapy (ddmm.mmmmN) 2004H/1					PozatekMapy (dddmm.mmmmE) 2004H/2			
	RPDO4 + MP	0x501	Výchozí Poloha X [mm] 2005H/1					Výchozí Poloha Y [mm] 2005H/2			
	RPDO1 + DM	0x205	OtackyMotoru [ot/min] 2010H/1		AktivaceBrzdy 2010H/2						
	TPDO1 + DM	0x185	AktualniOtackyMotoru [ot/min] 2006H/1		ProudMotoru [mA] 2006H/2		Brzda [0,1] 2006H/3	SmerOtaceni [0,1] 2006H/4			
	TPDO1 + MSp	0x182	Ultrazvuk [mm] 2007H/1		IR0 (levý) [mm] 2007H/2		IR1 (pravý) [mm] 2007H/3				
	TPDO2 + MSp	0x282	Osvětlení [-] 2008H/1	Nadproud [0,1] 2008H/2	Teplota [°C] 2008H/3						
	TPDO4 + MSp	0x482	f=0,addr=nodeID -	index -		subindex -	jeden z 64 různých objektů TPA81 2009H/(1-64)				
	TPDO1 + MSz	0x183	Ultrazvuk [mm] 200AH/1		IR2 (pravý) [mm] 200AH/2		IR3 (levý) [mm] 200AH/3				
	TPDO2 + MSz	0x283	Osvětlení [-] 200BH/1	Nadproud [0,1] 200BH/2	Teplota [°C] 200BH/3						
	TPDO1 + MP	0x181	Poloha X [mm] 200CH/1					Poloha Y [mm] 200CH/2			
	TPDO2 + MP	0x281	G - osa X [raw] 200DH/1	G - osa Y [raw] 200DH/2	G - osa Z [raw] 200DH/3	Plyn [%] 200DH/4	Tlak[kPa] 200DH/5	Natočení [°] 200DH/6			
	TPDO3 + MP	0x381	hhmmss\0 200EH/1								
	TPDO4 + MP	0x481	PolohaAuta (ddmm.mmmmN) 200FH/1					PolohaAuta (dddmm.mmmmE) 200FH/2			

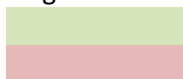
Zařízení	Zpráva	Identifikátor (COB-ID)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Modul pohonu	TPDO1 + DM	0x185	AktualniOtackyMotoru [ot/min] 2001H/1		ProudMotoru [mA] 2001H/2		Brzda [0,1] 2001H/3	SmerOtaceni [0,1] 2001H/4		
	RPDO1 + DM	0x205	OtackyMotoru [ot/min] 2002H/1		AktivaceBrzdy 2002H/2					
	TPDO1 + BM	0x184	NabijeciNapeti [mV] 2003H/1		NabijeciProud [mA] 2003H/2		NapetiBaterie [mV] 2003H/3		OdebiranyProud [mA] 2003H/4	

Zařízení	Zpráva	Identifikátor (COB-ID)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Modul světel - přední	TPDO1 + MSp	0x182	Ultrazvuk [mm] 2002H		IRO (levý) [mm] 2004H/1		IR1 (pravý) [mm] 2004H/2			
	TPDO2 + MSp	0x282	Osvětlení [-] 2005H	Nadproud [0,1] 2006H	Teplota [°C] 2007H					
	RPDO1 + MP	0x201	Ujetá dráha [mm] 2001H		SkutečNatočeníKol 200AH					
	TPDO4 + MSp	0x482	f=0,addr=nodeID	index		subindex	jeden z 62 různých objektů TPA81			
	RPDO1 + MSp	0x202	Tranzistory [3bit] 2008H	Natočení kol [°] 2003H						

Zařízení	Zpráva	Identifikátor (COB-ID)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Modul světel - zadní	TPDO1 + MSz	0x183	Ultrazvuk [mm] 2002H		IR2 (pravý) [mm] 2004H/2		IR3 (levý) [mm] 2004H/1			
	TPDO2 + MSz	0x283	Osvětlení [-] 2005H	Nadproud [0,1] 2006H	Teplota [°C] 2000H					
	RPDO2 + MP	0x302	Ujetá dráha [mm] 2001H							
	RPDO1 + MSz	0x203	Tranzistory [3bit] 2008H	Natočení kol [°] 2000H						

Zařízení	Zpráva	Identifikátor (COB-ID)	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Modul polohy	TPDO1 + MP	0x181	Poloha X [mm] 2005H/1				Poloha Y [mm] 2005H/2			
	TPDO2 + MP	0x281	G - osa X [raw] 2007H/1	G - osa Y [raw] 2007H/2	G - osa Z [raw] 2007H/3	Plyn [%] 2008H	Tlak [kPa] 2009H	Natočení [°] 200AH		
	TPDO3 + MP	0x381	hhmmss\0 200BH							
	TPDO4 + MP	0x481	PolohaAuta (ddmm.mmmmN) (šířka) 200CH/1				PolohaAuta (dddmm.mmmmE) (délka) 200CH/2			
	RPDO1 + MP	0x201	Ujetá dráha [mm] 2001H		SkutečNatočeníKol 200FH					
	RPDO2 + MP	0x302	Ujetá dráha [mm] 2002H							
	RPDO1 + MSp	0x202	Tranzistory [3bit] 2000H	Natočení kol [°] 2003H	VychoziNatoceniVozidla 200EH					
	TPDO1 + DM	0x185	AktualniOtackyMotoru [ot/min] 2000H		ProudMotoru [mA] 2000H		Brzda [0,1] 2000H	SmerOtaceni[0,1] 2004H		
	RPDO3 + MP	0x401	PocatekMapy (ddmm.mmmmN) (šířka) 200DH/1				PozatekMapy (dddmm.mmmmE) (délka) 200DH/2			
	RPDO4 + MP	0x501	Výchozí Poloha X [mm] 2006H/1				Výchozí Poloha Y [mm] 2006H/2			

Legenda



MSp	2	Vysíláno z modulu na síť
MSz	3	Přijímáno ze sítě do modulu
MP	1	Modul světel - přední
DM	5	Modul světel - zadní
BM	4	Modul pohony
		Modul správy napájení

VII. Fotodokumentace



Sestavený funkční prototyp zařízení



Ovládací panel (vizualizační stanice)

VIII. DVD-ROM

Obsah přiloženého DVD

Adresář	Obsah adresáře
\Dokumentace\	Obsahuje volně šiřitelné texty (dokumentace)
\Foto dokumentace\	Fotografie vozidla a jeho částí
\software\	Software pro moduly (Code Warrior)
\hardware\	Schémata zapojení a DPS (Formica)